U.S.A. $3.50
CANADA $4.50

Atari Computer Magazine

Antic®

The ATARI® Resource

JUNE 1985 VOLUME 4, NUMBER 2

COMPUTER ARTS

Atari, Casio & Yamaha Make Music Together

Generate 3-D Action

Weave Color on Atari

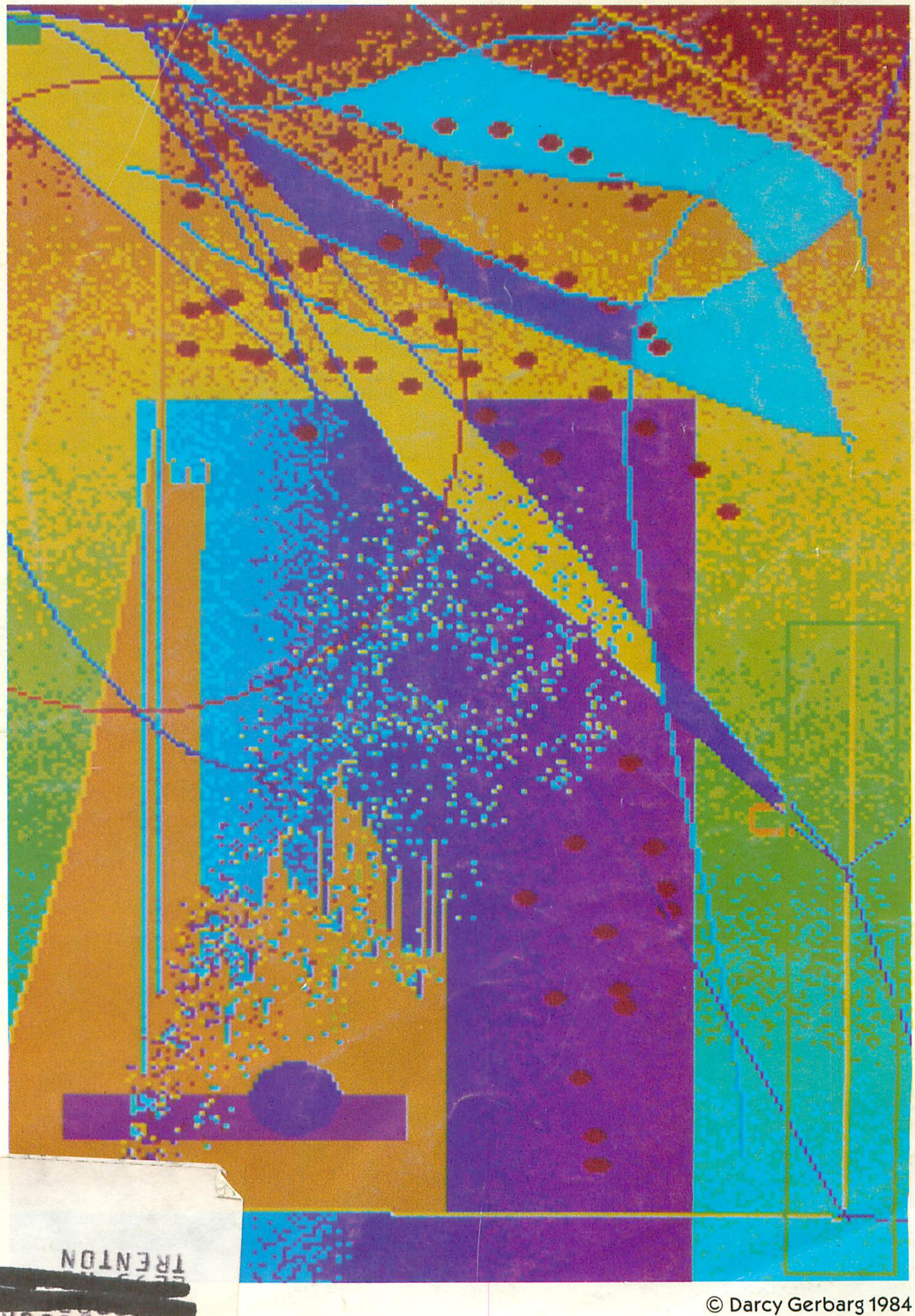
New Graphics Commands from BASIC

Cover Contest Winners

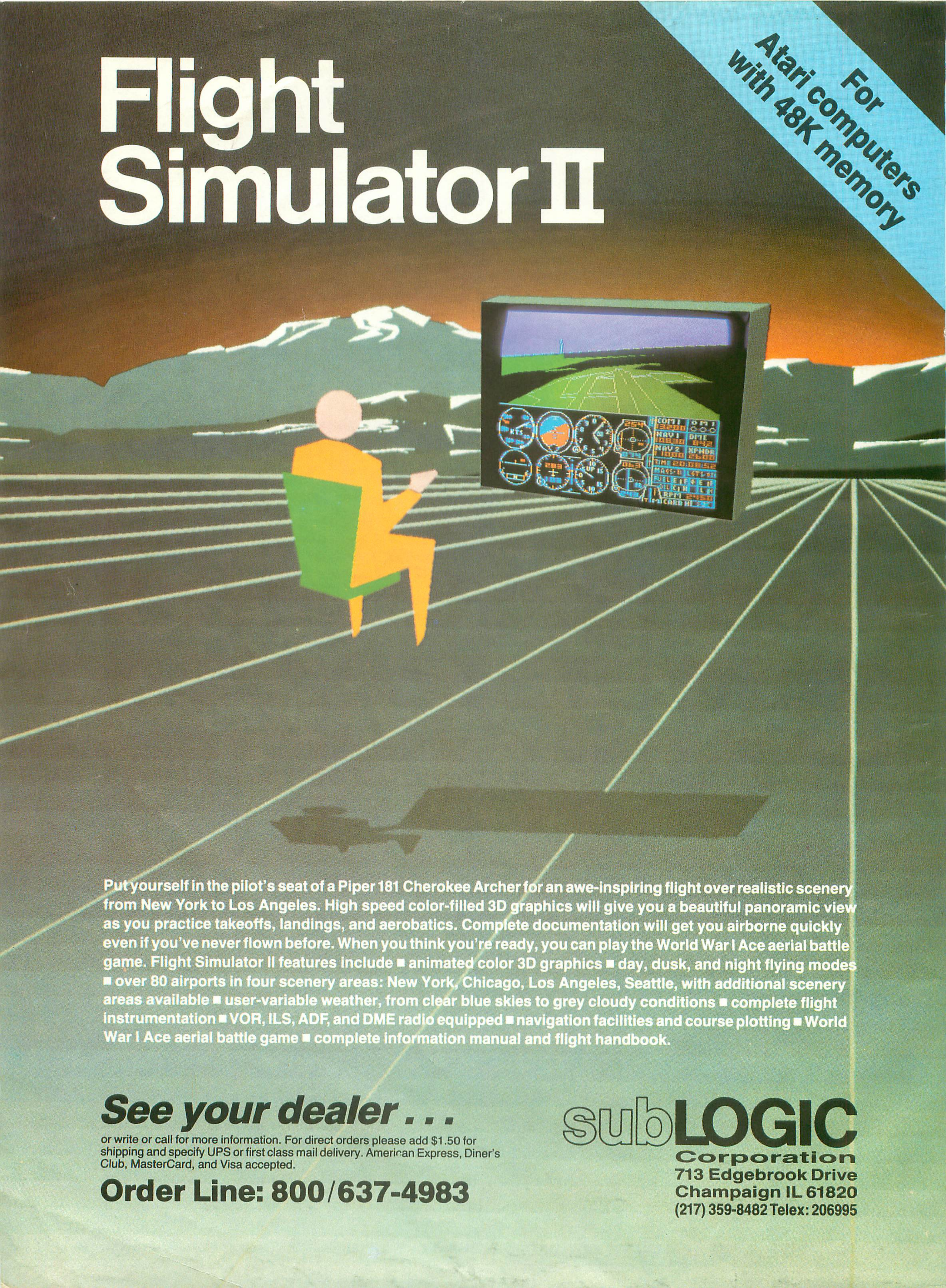
GEM Seminar ST Support

10 Type-In Programs

(25 Pages of Listings)



© Darcy Gerbarg 1984


Flight Simulator II
For Atari computers with 48K memory
Put yourself in the pilot's seat of a Piper 181 Cherokee Archer for an awe-inspiring flight over realistic scenery from New York to Los Angeles. High speed color-filled 3D graphics will give you a beautiful panoramic view as you practice takeoffs, landings, and aerobatics. Complete documentation will get you airborne quickly even if you've never flown before. When you think you're ready, you can play the World War I Ace aerial battle game. Flight Simulator II features include ■ animated color 3D graphics ■ day, dusk, and night flying modes ■ over 80 airports in four scenery areas: New York, Chicago, Los Angeles, Seattle, with additional scenery areas available ■ user-variable weather, from clear blue skies to grey cloudy conditions ■ complete flight instrumentation ■ VOR, ILS, ADF, and DME radio equipped ■ navigation facilities and course plotting ■ World War I Ace aerial battle game ■ complete information manual and flight handbook.
See your dealer . . .
or write or call for more information. For direct orders please add $1.50 for shipping and specify UPS or first class mail delivery. American Express, Diner's Club, MasterCard, and Visa accepted.
Order Line: 800/637-4983
subLOGIC
Corporation
713 Edgebrook Drive
Champaign IL 61820
(217) 359-8482 Telex: 206995

Join the Atari® Safari

of Software and Books



Get Out of the Thick of Things

If too many bills are piling up around you, *The Home Accountant*™ will put you back on safe ground. *The Home Accountant* handles cash, checkbooks and credit cards. Custom retrieval allows you to print all the information you need, including net worth, balance statements and checks.

Learn to Create Wild Graphics

Create your own electronic art with *Atari Color Graphics: A Beginner's Workbook*. So simple to follow, even a complete novice can program stunning graphic displays by following a few easy steps.

Program Ferocious Games

Atari Graphics and Arcade Game Design will teach you how to design quality games on the Atari. A series of flow charts, text and commented code explain everything in detail. Five complete games show you how it's done.

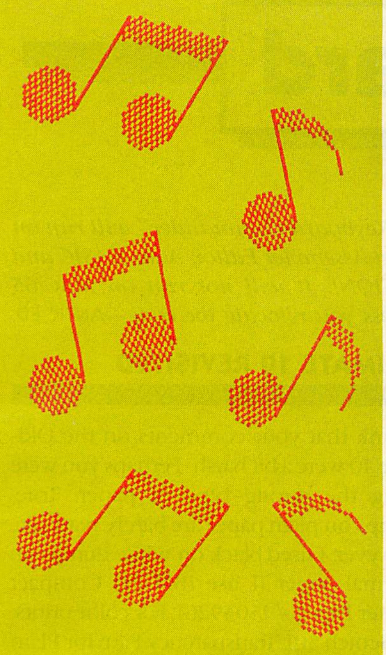
Conquer Your Computer

If you're wondering what to do with your Atari now that you've played countless games, get *The Atari User's Encyclopedia*. This book tells everything you need to know for creative and satisfying use of your computer.

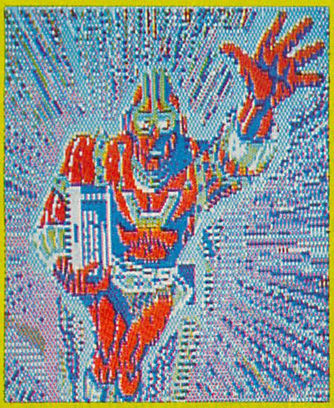
Put an end to the computer software and book hunt today. **Write or call our Product Information department for the dealer in your area.**

Arrays, Inc./Continental Software
Arrays, Inc./The Book Division
Attn: Product Information
11223 S. Hindry Ave., Los Angeles, CA 90045
(213) 410-3977

The Home Accountant hardware requirements: 48K Atari 800/800XL, one or two disk drives, printer (132 Col. optional)



A few notes on Print Shop 25



Color-the-Cover winners 22



Cows? Helicopters? You bet! 48

Antic®

The ATARI® Resource

JUNE 1985, VOLUME 4, NUMBER 2

FEATURES

DEPARTMENTS

SOFTWARE LIBRARY

Antic

Publisher
James Capparell

Editorial Department
Nat Friedland, Editor; Jack Powell, Technical Editor; Michael Ciraolo, Associate Editor; Charles Jackson, Program Editor; Ron Luks, On-Line Editor.

Contributing Editors
Carl Evans, Ken Harms, Jerry White, Suzi Subeck, Anita Malnig.

Art Department
Marni Tapscott, Art Director; Diane Lindley, Production Supervisor; Linda Tapscott, Ad Production Coordinator; Julianne Ososke, Production Assistant.

Cover Artist
Darcy Gerbarg

Contributing Artists
Rosiland Solomon, Beatrice Benjamin

Circulation Department
Les Torok, Manager; Hun-sik Kim, Peter Walsh, Shipping; Monica Burrell, Subscriptions; Eve Gowdey, Dealer Sales; Doug Millison, Dealer Sales; Brandt/Klingle, Circulation Consultants.

Accounting Department
V.J. Briggs, Manager; Brenda Oliver, Accounts Receivable; Lorene Kaatz, Credit Manager; Andrew Pope, Customer Service, Retailers.

Marketing
Gary Yost, Manager, Marketing Services; Steve Randall, Advertising Sales Director; Harvey Bernstein, Advertising Sales; Garland Associates, East Coast Representative.

Maria E. Chavez, Receptionist

General Offices (415) 957-0886
Advertising Sales (415) 661-3400
Credit Card Subscriptions
outside California (800) 227-1617 ext. 133
inside California (800) 772-3545 ext. 133
Subscription Problems (415) 957-0886

June 1985, Volume 4, Number 2
Antic—The Atari Resource is published twelve times per year by Antic Publishing. Editorial offices are located at 524 Second Street, San Francisco, CA 94107. ISSN 0745-2527. Second Class Postage paid at San Francisco, California and additional mailing offices. POSTMASTER: Send address change to **Antic**, 524 Second Street, San Francisco, CA 94107.

Editorial submissions should include program listing on disk or cassette, and text file on media and paper if text was prepared with a word processor. Media will be returned if self-addressed stamped mailer is supplied. **Antic** assumes no responsibility for unsolicited editorial material.

i/o board

BEATING THE ZURK BLUES

I've been having problems typing in line 5000 of "Zurk" (March 1985). Please help.

Douglas Hiatt
Largo, FL

*Line 5000 contains the phrase "SET1=L1*256". We are going to assume here that you're using the BASIC XL cartridge from O.S.S. This BASIC is more powerful than Atari BASIC and has several extra commands. One of these commands is SET and thus, it cannot be used as a variable without special precautions. See page 131 of your BASIC XL manual on this. "ZURK" runs as published and should work in BASIC XL if you change all the SETs to some other variable name.*—ANTIC ED

TYPO II TYPO

TYPO II (January 1985) does not work on my 1200 XL. It doesn't accept the SET commands. Any suggestions?

Mary Mitcham
Houston, TX

*The original instructions in "How to use TYPO II" read "BASIC XL **cartridge** owners type SET 5,0", etc. We have subsequently cleared up the wording.*
—ANTIC ED

SOME THOUGHTS ON AL

In most issues of Antic, there are type-in programs listed in BASIC as well as in assembly language. I've spent many hours typing in AL programs and found that none worked. In particular, I've had problems with "Keyboard Commander", (March 1985). Can you help me?

Henry Jennings
Detroit, MI

If the AL listing is an addendum to a BASIC program, it's printed primarily as a study example for serious AL programmers. Unlike BASIC, machine language is unforgiving. One mistyped bit of code and your computer will lock up.

"Keyboard Commander" loads into Page Six. So does MAC/65. Unfortunately, there is only room for one program in Page Six, and the big orange supercartridge is not about to let it be "Keyboard Commander".

"Keyboard Commander" will run on Atari Assembler Editor, Atari BASIC and ACTION!. It will not run on MAC/65 unless you relocate the code.—ANTIC ED

OKIMATE 10 REVISITED

I think that your comments on the Okimate 10 were a bit harsh. Perhaps you were using the wrong type of paper. True, dumps on plain paper are barely readable. However, I used black on white dumps on thermal paper (I use IBM PC Compact Printer Paper, # 1503926). For color prints, try Scotch 501 Transparency Film for Plain Paper Copiers.

Alan Fillmore
Bakersfield, CA

*Several readers have suggested that we used the wrong type of paper in reviewing the Okimate 10. Well, **Antic** finds it refreshing to be considered too tough on an advertiser for a change. We recognize that some readers have had better results with that printer than we did.*

When we reviewed the Okimate 10, we used the paper, printhead, printer and ribbons provided by the Okimate Corporation. We followed the company's instructions—but then wound up spending most of a workday tinkering with the configuration in order to get even the slightly improved results we printed.

We assume that a major company would ship a working, pre-tested piece of equipment to assure the best possible review. If we were, in fact, provided with a bad printhead that we didn't recognize at the time, this is a noteworthy problem that might be faced by anybody purchasing the printer.—ANTIC ED

OF BITS AND BYTES

What is so special about the number 256, as in 10 PEEK A(195)*256?

Kevin A. Scott
Algonquin, IL

The highest number you can have in any one address is 256. Your Atari is an eight-bit, binary computer. "Binary" means it only understands two numbers, 0 and 1. These numbers are called "bits". Eight bits make a "byte" and the maximum different combinations of eight bits is 256 (0–255).

i/o board

Since memory is nothing but a sequence of byte addresses, and we want to reach more than 256 of them, we put two bytes together and have 256 combinations times 256, or, 65,536.

These double-byte numbers are called "words" and the two bytes that make up a word are called the "low-byte" and the "high-byte". When the low-byte goes beyond 255 (remember 0–255) it returns to zero and the high-byte is upped by one. This means that each unit in the high byte is equal to 256.

*Now if, for example, the high-byte contains 2, its value is 2*256, or 512.*

*And if the low-byte contains 50, the two bytes together equal 562. Words are stored in the Atari in a backwards order of low-byte followed by high-byte. If the number 562 was stored in locations 88 and 89, 88 would hold 50 and 89 would hold 2. The formula to find the values of these two locations is: WORD=PEEK(88)+PEEK(89)*256*—ANTIC ED

BATTERIES MOVES

Batteries Included, publishers of PaperClip (reviewed in **Antic** last month) moved to 30 Mural Street, Richmond Hill, Ontario, L4B 1B5 Canada. However, please don't order products from them by mail.

help!

KWIK DUMP

"Kwik Dump" (**Antic**, March 1985) contains an error in line 1070. The last number in that line should be 27 instead of a zero.

THIEF

Readers are having problems typing in "Thief", the March 1985 Game of the Month. The game does run as published, but if your TYPO II code for line 1105 is UK, make sure that the inverse [p] in that line is lower-case.

WIDE TEXT

Bill Morris's "Wide Text" (**Antic**, January 1985) cannot print a double-width [Z]. Substitute these lines to correct this problem:

In the assembler listing:

```
0600        CPY #27*8    ;8
 BYTES TO A LETTER
```

In the BASIC listing:

```
CT 20130 DATA 216,208,2
13,185,0,224
```

S.A.M. SOURCE CODE

The assembly language listing of "S.A.M. Handler" (**Antic**, February 1985) contains an error in line 900. LDY $01 should read: LDY #01. The BASIC listing is correct, however, and runs as published.

TYPO II BONUS

Ever since our improved TYPO II program typing proofreader began appearing in January, **Antic** has been getting letters that ask for TYPO II codes to the most popular and difficult pre-1985 programs. These will be in the issue Next Month!

You'll find the TYPO II line codes for the four most-requested listings—"Biffdrop," "Escape From Epsilon," "Advent X-5" and "Adventure Island." And we'll print codes for more **Antic** golden oldies if we get enough requests.

CHECKED 100 TIMES

I typed in a program from the July 1984 issue and it still doesn't run. I checked my listing 100 times and I have not found a typing error. I also checked the following issues for an error report but didn't find one.

H. Reynaldos
Miami, FL

This is typical of many letters we receive from frustrated readers who type in a program only to find it does not run. They may check it many times and when it still won't work, they naturally suspect a publishing error.

***Antic** does make mistakes, as you can see from the magazine's **Help!** items and the monthly **Error File**. But the great majority of program problems come from readers mistyping the listing or misunderstanding the instructions.*

*If you are having problems with a program that was published more than three months ago, and you don't see a correction printed in either **Help!** or **Error File**, you can bet the error is on your end.*

*That's because most major program problems are discovered within two weeks of publication! But due to the long turnaround time for four-color national magazines, the soonest you can expect that error to appear in **Help!** is two issues later. For example, a February error should show up in April's **Help!***

*Program errors which appear in **Help!** are transferred the following month to the **Error File** where they remain for approximately six months. A complete index of all **Antic** errors can be found on our ANTIC ONLINE service on CompuServe.*

We test all programs on Atari 800s, 1200XLs, 600XLs and 800XLs using both cartridge and built-in BASIC. We use Atari 810, 1050 and Indus GT drives with Atari DOS 2.0S, single density.

Some of our listings are more difficult to type in than others. If you are just starting, we strongly suggest that you first try the shorter listings and avoid the frustration of typing in very long listings with Atari special characters.

*When using TYPO II, be careful that you have typed in **every** line. It's not so hard to miss one entire line and TYPO II won't spot it. If you're sure all the lines are there and the program still won't run, recheck each TYPO II line code.*—ANTIC ED

THE #1 SYSOP

*Ron Luks of SIG*ATARI*

by MICHAEL CIRAOLO,
Antic Associate Editor

Back when the IBM PC had just come out and Ron Luks was thinking about buying one, a friend showed him Atari's classic Star Raiders game. Luks bought an Atari to play with while he was on the IBM waiting list. As it turned out, he never bought an IBM PC.

Instead Luks, 33, became the system operator (sysop) of SIG*Atari on CompuServe—the world's largest Atari bulletin board, with some 6,000 enrolled members.

At the time he bought his Atari, Luks was a stockbroker and money manager who traded options on the American Stock Exchange. Previously he had taught scuba diving in Miami for a few years. "Fooling around with the Atari was light and fun, like a puzzle, after a high-pressure day in Wall Street," he said.

One of Luks's first peripherals was a modem. "I got the Hayes SmartModem, an 850 interface and the original TeleLink cartridge, which came with a CompuServe Starter Kit."

Once you read this latest issue of **Antic**, log onto CompuServe and type GO ANTIC—you'll see a preview of the magazine's NEXT issue!!

In the ANTIC ONLINE preview, you'll find a comprehensive look at all the stories and programs in the next **Antic** Magazine—the July Computer Challenges issue.

You'll even find a major excerpt or two from the upcoming issue's featured articles—such as our Atari chess software tournament or our interview with the boss of Strategic Simulations, Inc.

And as a special CompuServe bonus, *you can now download from SIG*Atari the complete program that won **Antic's** Color-The-Cover Contest*. This scrolling two-screen picture won't be on the monthly Antic Disk until next issue, even though you'll find its photo in the current magazine. To download this file, type DL4 at the SIG*Atari prompt. This puts you into Data Library 4, where BRO WINNER.* is what you type next. If you're using an Xmodem protocol program (such as HomePak or Chameleon) choose the WINNER.XMO file. With TSCOPE, use the WINNER.BIN file.

ANTIC ONLINE special bulletins may be downloaded for reprinting in newsletters of users groups affiliated with the Antic Worldwide Users Network. Officers of Atari users groups may write to the **Antic** WUN Coordination for details.

EARLY COMPUSERVE

Back then, CompuServe wasn't primarily concerned with sponsoring special interest groups (SIGs). So when Luks first logged onto the system, he "looked all over for the word Atari." It wasn't there. Luks then began to spend time in the Popular Electronics magazine online edition.

"In those days, the network had no online sysops, no help. It was like a big puzzle," Luks said. "I kept leaving messages—How can I do this? Why can't it do that?" The messages were picked up by the sysop about once a month.

CompuServe finally collapsed under Luks' badgering. "They made me an assistant sysop and gave me a free flag," which meant he would have free access to the network. This was a good deal for a man who had monthly connect bills "approaching four figures".

FILLING PCS-132

One thing led to another, and Luks was soon allowed to set up an Atari board. "They said 'There are some empty pages at PCS-132' and I got to fill them up." Luks took as assistant sysops a few of the most active members interested in Atari—such as Michael Reichmann of Batteries Included and programmers Steve Ahlstrom (SynFile+, PaperClip) and Russ Wetmore (Preppie, HomePak).

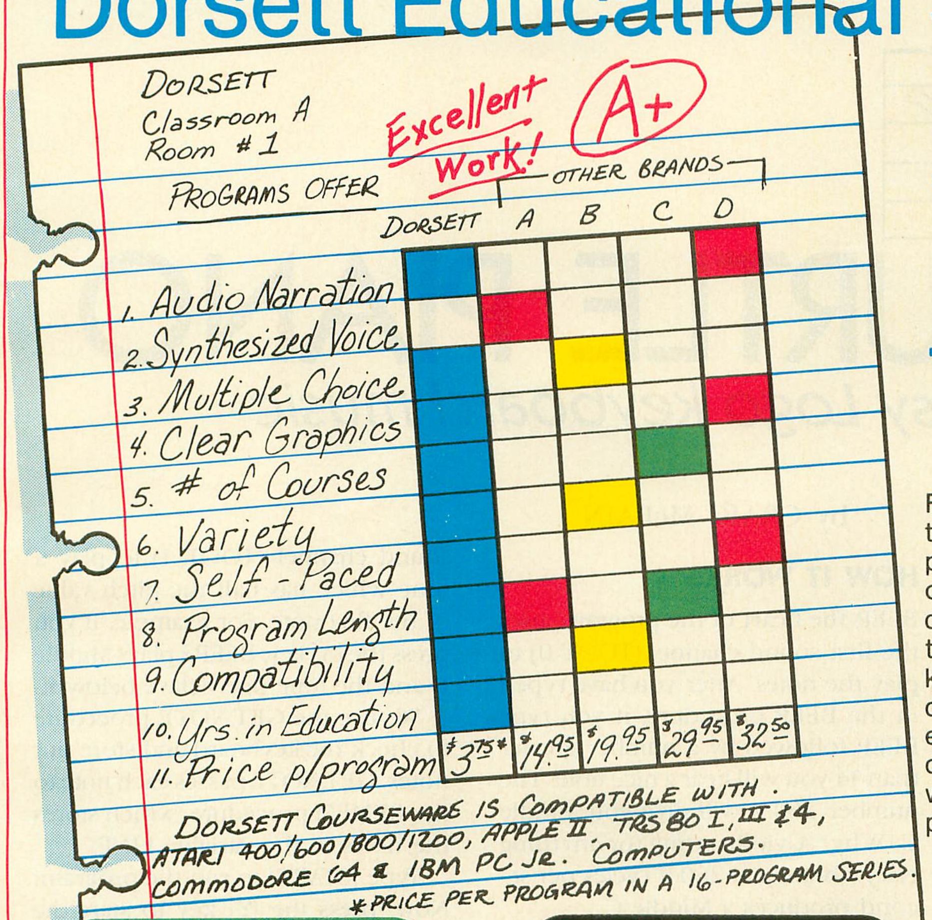
After an all-nighter over cappuccino in Greenwich Village, Luks came up with the name SIG*Atari. "It's different from 'the Atari SIG,' the 'Radio Shack SIG' and so on. Atari is a word

continued on page 19

For Your Atari Computer

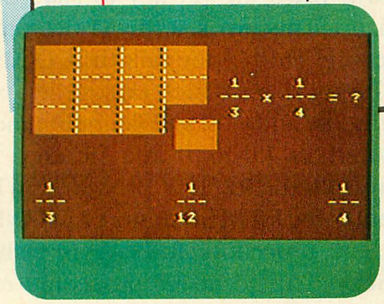
TRS 80 Color I, III & 4 • Apple • Commodore • IBM PC Jr.

Dorsett Educational Software Passes the Cost-Efficiency Test!

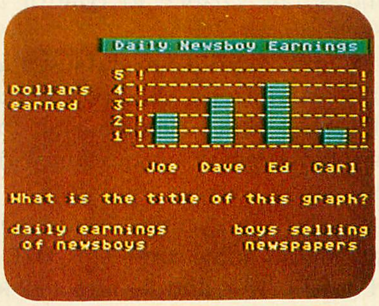


For computer assisted instruction, our software more than passes the test...it EXCELS! With over 1000 educational programs designed specifically to concentrate learning for all users, from kindergarten level to advanced individual studies, from slow learners to exceptionally bright students, you'll get positive results with our quality, economically-priced courseware.

• 500 Programs Available for TRS 80 Color; 400 for IBM PC Jr.







Interactive Tutorial Programs

Over 1000 Programs with Full Time Audio Narration, Pictures & Text!

We're Your Educational Software Source

Course	No. of Programs
Reading	256
Mathematics	128
Comprehension	48
History	32
Algebra	16
Spelling	16
Government	16

16 Programs in each of the following:

Carpentry - Electronics - Health Services
Office Skills - Statistics - First Aid/Safety
Economics - Business - Accounting
Psychology

AND MANY MORE!

Dealer inquiries welcome

Atari 400/600/800/1200 computers require the Atari cassette recorder and the Dorsett 4001 Educational Master Cartridge, **$9.95.** Apple II, TRS 80 I, III, & 4, and Commodore 64 computers require respective conversion kits (plug-in board and stereo cassette player), **$99.00.** For the IBM PC Jr. a cassette adapter cable and a good cassette recorder are required. A Radio Shack CCR-81 or CCR-82 is recommended. **$59.90** for an album containing a 16-program course (8 cassettes with 2 programs, each at **$3.75** per program). **$8.80** for a 2-program cassette.

Send for a catalog of over 1000 programs for Atari, TRS 80, Apple, etc.

For more information, or to order call:

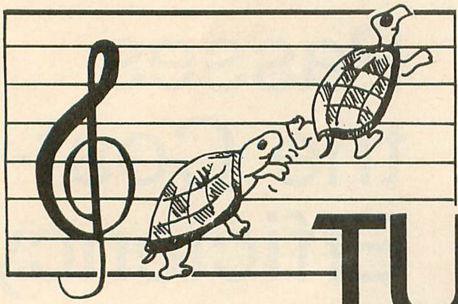
TOLL FREE 1-800-654-3871

IN OKLAHOMA CALL (405) 288-2301

DORSETT
Educational Systems, Inc.

Box 1226 Dept A Norman, OK. 73070

MasterCard VISA



TURTLE PIANO

Easy Logo keyboard music

by CRAIG McBAIN

A simple but adaptable Logo program that plays a musical scale from the Atari keyboard. Works on all Atari computers of any memory size. Disk or cassette.

Children love to sit in front of a piano keyboard and pound out simple melodies. The child's joy comes from immediate success at making music and a rare chance to control the environment.

With an Atari, Logo and just a few short commands, children can have the same kind of musical fun they would have with a piano, but with a little extra learning thrown in.

Turtle Piano converts the computer keyboard's home row—A, S, D, F, G, H, J, K, L, ;, + and *—into a 12-key piano that plays a do-re-me scale in the Key of C without sharps and flats.

It also changes the screen turtle into an eighth note which jumps around to the appropriate spot on a musical staff whenever a note is played. As each note is played, its letter-name is also printed on the screen.

Turtle Piano can also remember your compositions so you may listen to them over and over again.

Though it is not a particularly complex program, Turtle Piano is an engrossing project for young computer users and programmers. So type in Listing 1 and SAVE a copy before you RUN it.

HOW IT WORKS

BEEP, the heart of the program, uses the first sound channel (TOOT 0) to play the notes. After you have typed in the BEEP procedure, if you type BEEP followed by a number greater than 14 you will hear a nice note. The number 260.7 will give you Middle C. Why? A violin string (or anything else) vibrating at 260.7 cycles per second produces a Middle C.

The PLAY procedure ties the home row keys to their assigned pitch values. If a key is pressed which is not one of the home row keys, PLAY will output a frequency of 50000 (too high to hear). So you can use the space bar, for example, as a rest between notes. PLAY, using the SETY command, also draws the turtle on the staff.

Type BEEP PLAY "A to hear Middle C. PLAY takes the input character [A] and searches for its corresponding pitch value (260.7). PLAY passes this value to BEEP, which plays the appropriate note.

Note the relationship between the values for Middle C (260.7) and High C (521.5) in the PLAY procedure. Now, look at the values for the E and F tones. The higher note in each pair is double the pitch value of the lower note. This is always true. BEEP2, the next procedure, uses this fact to produce interesting results.

Try BEEP2 a few times and compare it with BEEP. It uses the second sound channel (TOOT 1) to play a tone which has half the pitch value of the first note. For example, if you press the [A] key, BEEP2 plays Middle C and the tone one octave below it.

We use the GET.NOTE procedure to check the keyboard and store our notes. GET.NOTE passes each note to the REMEM procedure, which stores the notes in the variable LINE.

Type [START] to run the program. Now, press the [Y] key to start the piano. This invokes the START procedure. The [C] key calls the CLEANUP procedure which erases the text window and erases all your old tunes stored in LINE. If the [R] key is pressed, the REPLAY procedure will replay the list of notes stored in LINE.

The procedure REPLAY is really tricky. This recursive procedure removes the top note from LINE, plays it, and discards it. Then, it starts again, playing the first note on the shortened list. REPLAY continues to remove notes from the top of LINE, play them and discard them until LINE is empty.

The graphics portions of the program need little comment except perhaps for the LOGO names "STAFFHI, "STAFFLO, and "NOAT. Don't forget to type these, too. These variables contain lists of numbers which tell the computer how to reshape the turtles into musical notes. The turtles are reshaped in the SETUP procedure with the PUTSH command.

continued on page 19

ANTIC AT THE FIRST GEM SEMINAR

Converting IBM PC software to Atari

by NAT FRIEDLAND, **Antic** Editor

Antic was the ONLY Atari Magazine present at the first GEM Software Developers Seminar held at Monterey, California in February by Digital Research, Inc. DRI created the "Macintosh-like" GEM operating interface that is being used for the new 16-bit Atari ST computers.

GEM has aroused especially wide interest in the computer industry because it's supposed to make it simple to convert software between the IBM PC and clones, the Macintosh, the Atari ST and any other computer that GEM licenses a version for.

Some of Atari's top technical executives were on hand, debuting a working ST with a preliminary version of Atari GEM burned into ROM.

Antic was told that Atari still considers itself on schedule for bringing the first production ST computers onto the market in April. Full ST production capacity won't be reached until June. The 10–15 megabyte hard disk for the ST will show up in the summer.

ATARI CAD/CAM?

Remember the 32-bit Atari we reported Jack Tramiel talking about at his November press conference? Well, apparently it is well along in development. Atari still hopes to meet Tramiel's goal of unveiling the machine at the April electronics fair in Hanover, Germany.

Every time Atari engineers talked about the 32-bit computer in Monterey, delighted smiles appeared on their faces. The computer was described to **Antic** as a "VAX minicomputer on a chip" and a "$40,000 CAD/CAM computer graphics workstation that will sell for under $2,000."

LEARNING ABOUT GEM

As for the GEM Seminar itself, the $800 workshop was highly technical and directed at professional consumer-software programmers who were thoroughly experienced with the C language or with Macintosh window program development.

The Seminar sessions were taken up with highly detailed discussions of GEM development nuts and bolts such as the strict interfacing procedures which are supposed to make "porting" GEM-based programs between different computers a routine one-day process.

Access to GEM windows, debugging, and correct embedding of transfer hooks were among the other technical topics discussed. All attendees were given the two-volume GEM Toolkit documentation. DRI's recommended professional development language was Lattice C, which costs around $500.

It should be noted that the Seminar was specifically dealing with the just-completed IBM PC version of GEM. Six-disk beta test editions of GEM were being sold to developers by DRI for $500. The GEM Library software of prepared graphics routines cost extra.

The Atari version of GEM was not yet ready for beta testing at the time of the Seminar.

WETMORE SUMS UP

Russ Wetmore, author of **Homepak** and **Preppie** as well as other major Atari programs, flew in from his Florida homebase to attend the seminar and then stayed on to see the MacWorld Show in San Francisco.

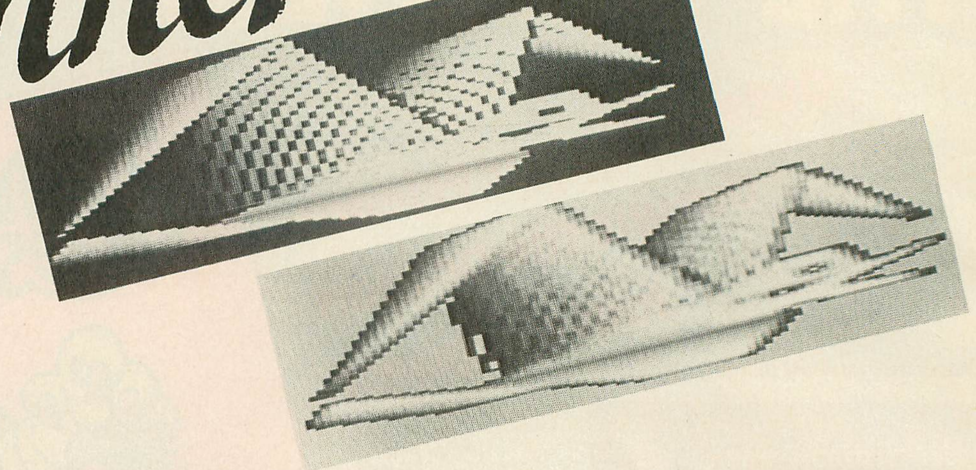
He spent time at **Antic** during this period and shared with us the viewpoints of a highly experienced Atari professional programmer.

"I think the developers at the Seminar fell into two groups. One group is totally sold on the GEM goal of making a lot of different computers compatible with each other," said Wetmore. "And for now they are willing to overlook any unanswered questions that came up during the sessions. The second group is taking more of a wait-and-see attitude."

Wetmore expresses some doubt that GEM software will port between different computers as easily as DRI says it will. He also feels that GEM lacks certain built-in features found in the Macintosh interface—such as a text editor—which will make it more

continued on page 19

Pixel Scanner



by LYN BUCHANAN

This BASIC program introduces you to pixel averaging—an advanced graphics technique similar to the method NASA uses for enhancing satellite photo details. Pixel Scanner works on all Atari computers of any memory size. Disk or cassette.

The Atari's graphics capabilities are the best you can get for anywhere near the price. The problem most of us run into is that our pictures LOOK as if we did them ourselves.

This doesn't need to be the case, however. Pixel Scanner will perform automatic computer enhancement of a picture by using a process called pixel averaging, which gives a depth and fullness to an otherwise flat picture.

And that isn't all. By tinkering with this program a little, you can simulate the sophisticated computer enhancements used by NASA to bring out small details in satellite photos.

To get started, type in Listing 1, check it with TYPO II, and SAVE a copy.

The program includes a demonstration picture of a stick figure bird in Graphics 9 (lines 200–270), and the pixel averaging routine (lines 100–161). The picture was made using simple PLOT and DRAWTO commands; if you're enterprising, you can create your own picture by changing these commands in the demonstration picture.

The stick figure is drawn at the top half of the screen, leaving room for the enhanced version at the bottom.

HOW IT WORKS

After the picture is drawn, the program looks at each pixel and its surrounding pixels, then places their values into variables. This is done in lines 121–129 by means of the LOCATE command, which positions the cursor at a specified x,y coordinate on the screen and assigns the value of the byte for that data to a specified variable.

The format for a LOCATE command is:

LOCATE (x-coord),(y-coord),(variable)

The program then adds the values of all the variables together, and divides by the number of pixels sampled (line 131). Using that average value, it then re-colors the center pixel, to make it blend more smoothly with its surroundings (line 141).

WHAT'S A PIXEL?

The word "pixel" is computer jargon for "picture element." It stands for the smallest controllable element of a screen display.

Naturally, pixel size is determined by computer and video resolution limits, as well as by graphics mode selection. In Atari Graphics 8 mode, for example, a pixel is one scan line high by one-half color clock wide—which essentially matches the resolution of a home television receiver.

PIXEL SEPARATING

The Pixel Scanner demo program blends pixels. It could just as easily widen the difference between pixels—as is done in satellite photos to distinguish between extremely small variations.

To do this, change the way the value of K is figured. In line 131, K is figured as the average of all the variables. It would be just as easy to make K vary by, say, 4 times the average, if you change the command COLOR K (line 141) into COLOR K*4.

With a little experimentation, you will begin to find a wide variety of things to do with this routine. Other effects can be achieved by sampling only certain adjacent pixels and not others.

You might also try changing to Graphics 11 (line 201). You'll be surprised at the beautiful blends of colors which are produced.

Lyn Buchanan is a civilian programmer at Fort Meade, Maryland. He is also a programming and systems analysis instructor.

Listing on page 58

MORE REAL POWER FOR YOUR ATARI

DESK SET $39.00

Complete Desk Package

CALENDAR

CALENDAR is a perpetual calendar, an appointment calendar and also a card file. The perpetual calendar is a calendar of every month, past, present or future. The appointment calendar allows up to 15 entries to be made each day.

CARD FILE

The card file is a mail list program which holds up to 200 addresses. The printing format of card file includes continuous lists, labels or envelopes. Files can be printed; all the files from one file number to another; by zip code; by state or by selected files.

LETTER WRITER

LETTER WRITER is a preformatted letter writing program. LETTER WRITER can be used for any number of applications involving entering, editing and printing text. LETTER WRITER is designed to be easy to use and does not require extensive training. While LETTER WRITER is not a full word processing system, it performs 90% of the functions used by harder to use and more expensive word processors. DESK SET also contains a program that allows you to combine Card File and Letter Writer for interaction.

FINANCIAL CALCULATOR

FINANCIAL CALCULATOR answers virtually any questions concerning the cost of money, loans, and interest earned on savings, loans and investments. Plus, this program will give a complete interest earned table and amortization table. This program is a must for anyone serious about money.

FORECASTER

Forecast future events based on past information. Forecast profits, costs, sales trends, prices test scores, virtually anything. Edit, save on disk and test various elements to determine the outcome. FORECASTER is a powerful "what if" program – a must for business.

Two drive - double density - 48K required.

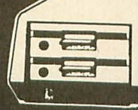
HOME WRITER $19.00

The HOME WRITER is an easy to use word processor which includes a carefully selected group of functions that are at your disposal immediately. The functions are as follows: SAVE, LOAD, REVIEW, PRINTOUT, or EDIT. All the popular editting features available on the ATARI Home Computer in direct programming mode are also available with HOME WRITER. You may also search for phrases or strings, load off of either disk or cassette, and number pages automatically. Unlike other small word processing programs, HOME WRITER does not wrap-around when at the end of a line. Right and left margin justification is available for any type parallel printer. 48K.

ASTRA

Double Disk Drives

True Double Density



Total Compatible

INCLUDES DOS

Big "D" Double Sided	$595.00
2001 Two Drive	$495.00
1620 Two Drive	$349.00

1620 quantities are limited.

SMARTDOS $39.00

- SMARTDOS is 100% density smart. SMARTDOS will sense the density of each disk in use and automatically reconfigure the entire system to that density.
- SMARTDOS does not require that a "system disk" has to remain in the drive, or be continually inserted and removed in order to use the DUP.SYS commands.
- With SMARTDOS you may Copy with query, (eliminates specifying each item individually).
- Counter screens – which keeps the user informed as to what the system is doing and where in the task the system is.
- Disk testing for bad or unusual sectors that may be corrected.
- RESIDUP feature allows simple yet powerful full time availability of DUP.SYS commands while leaving your program intact and ready to RUN.
- Minimum keystrokes for maximum power, e.g. a disk directory is done by pressing only one key – the drive number (great for filesearches, and "=" may be used to replace *.*.
- The ability to run from 1 to 9 autorun files sequentially.
- Built in disk drive speed check.
- SMARTDOS is only 34 single density sectors long and works with all Atari computers with a minimum of 24K RAM.

FILING SYSTEM $19.00

FILING SYSTEM allows the user to configure any type of data file imaginable. Examples are recipe cards, mail lists, reminders for birthdays, check-ups, etc..., complete inventories (home and business), personnel files, customer call-ups, price list, and much, much more. You may retrieve data using any field or combination of fields. Files also may be saved, sorted, and printed in a preset format that you configure. Uses either a single or a double density disk drive. 24K minimum.

LETTER SYSTEM $29.00

This disk contains both LETTER WRITER and MASTER LIST plus the necessary program that allows for combining the two programs for interaction. Call a file off disk, type a letter and (with the right printer) type on envelope, or just run one letter after another. All of the powerful features of both LETTER WRITER and MASTER LIST plus the ability to use them together.

Includes 35 pre-typed letters.

BILLING/INVENTORY $49.00

BILLING allows you to do invoices, posts to account cards, deductions from inventory and payments. There are provisions for shipping, discounts, no charge items and tax. BILLING also allows maintenance for up to 500 customer files, and printing of the daily total of billings. A flowchart is given to show the step-by-step direction that the program takes. Allows for up to 300 inventory items. 48K.

STAT-PLUS $29.00

STAT PLUS is the most powerful statistic tool we have seen. Run sample space, mean, variance and standard deviation. Do probabilities using binomial, poisson or normal distribution. Also, do students' t-test, Mann-Whitney U test and Chi Square. Do linear regressions; may use 1-12 independent variables, interact a sample from one module to another, edit, rank and print out. 24K min.

ACCESSORIES

Disk Doubler	$14.00
5' I/O Cable	$12.50
1' I/O Cable	$ 8.50
Transformer	$18.50
IBX 10 SS/DD Disks w/ plastic box	$14.50
C-5 Data Cassette w/box	$ 2.49
Disk Fix and Cleaning Kit	$29.00

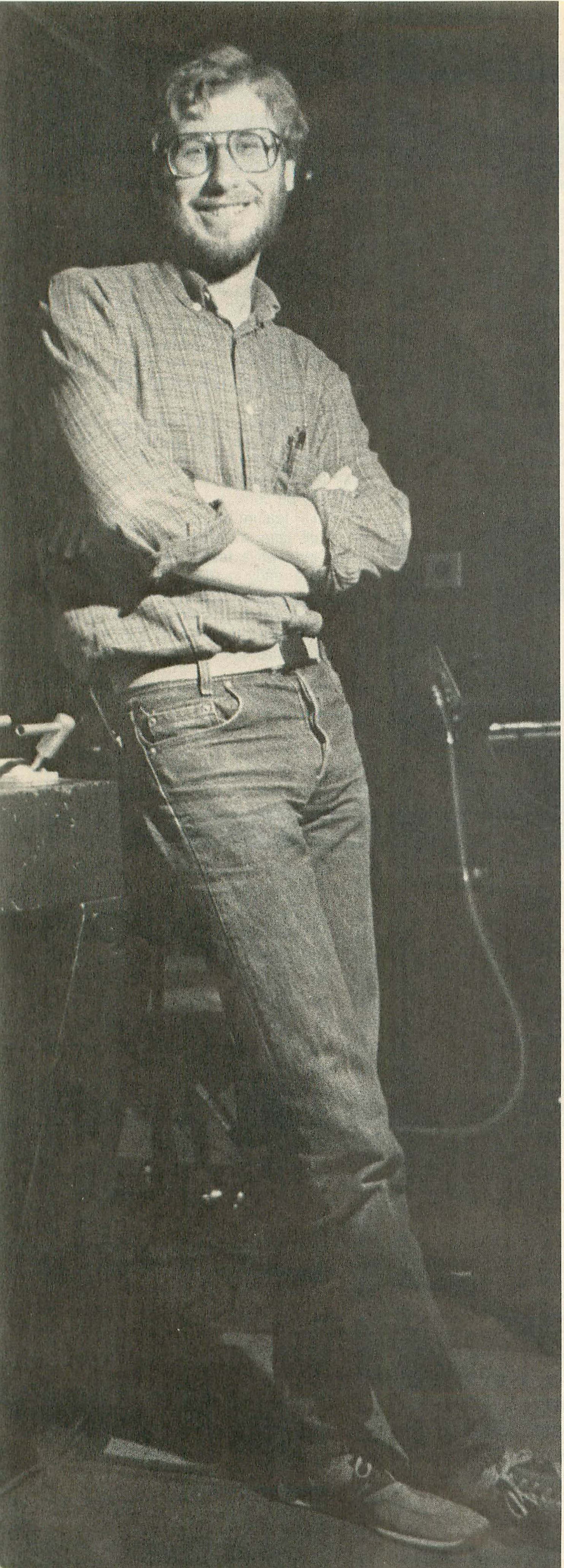
DEMONS DUNGEON $19.00

Queen Isidore's crown was stolen by the Demons that live in the deep terrifying dungeon. If the crown is not recovered the magic that protects her and the kingdom will vanish. Others have tried and died. Whoever recovers the crown and saves the kingdom will win the hand of the princess Ilena and the Kingdom.

THE PROGRAMMERS WORKSHOP

24686 REDLANDS BLVD. • SAN BERNARDINO, CA 92406
OUTSIDE OF CALIFORNIA **(800) 228-6821**
CALIFORNIA **(714) 796-6821**

TO ORDER: VISA/MASTERCARD, Check or Money Order accepted. If charge, please include expiration date. Shipping and Handling – Software $1.50 – Disk Drives $10.00. California residents add sales tax. Phone or mail.



Atari in Lights

Theater lighting designer's Atari C.A.D.

by MICHAEL CIRAOLO,
Antic Associate Editor

James Brentano is the stage lighting designer and lecturer for the Drama Department of U.C. Berkeley. He has also been a dedicated Atari hacker since he bought his 800 back in the days when it cost $800.

When you think about it, designing the lighting for a theatrical production is a highly information-intensive task. You have to combine dozens of color filters, lights, cables and dimmers in order to get the job done.

"A typical show will require 150–200 lights. For each light you must keep track of the following—name or label, any one of several hundred colors, circuit which plugs the unit into the control board, a dimmer switch and stage focus spot where the light is aimed. Also each light unit will have different level settings for up to 200 cues," Brentano said.

Designing the lighting for a show traditionally requires extensive drafting of diagrams and long lists of each light's location, focus and so on. For each show, lights, cables and other material also have to be ordered—more paperwork.

Brentano now uses his Atari to handle the entire process. He wrote a BASIC routine to draw the lighting characters in Graphic 8, and uses **Graphics Master** software to produce a design layout template.

He also uses **SynFile+** to keep track of lighting information which can be sorted—by focus, type of light, etc.

COMPUTER AIDED DESIGN

Brentano isn't content to simply turn the paperwork over to his Atari. He's planning to unite his lighting programs and utilities in one system, a computer aided design (CAD) package for the Atari.

"Nobody has yet developed a system where you're simultaneously hooked into the stage cues and all the lighting unit information," Brentano said.

"For the price of two IBM graphics boards and a whistle, I can produce a package with an Atari, dot-matrix printer, disk drives and the software to do *all* the CAD lighting work," Brentano figured. All for around $1,000.

"Ideally you'd have onscreen a picture of the lighting diagram, and you'd use a light pen or mouse to circle and call up all the information on given unit."

After producing a package to do CAD lighting design, the next step is direct computer control of the lighting board. "The technology of light board computers is primitive—the interface to all the knobs and controls is expensive," the designer explained.

The light board Brentano uses at U.C. Berkeley costs $37,000, not including the dimmers. "That's ridiculous when an Atari costs $120," Brentano said.

"LET'S MAKE A SHOW"

"It's surprising how many stagehands



LINDA TAPSCOTT

own Ataris—they like to play games. And we all play this game together, "Let's go make a show". Backstage crafts simply offer bigger and more expensive toys."

Brentano maintains that the Atari appeals to stagehands because of the machine's game tradition and because it's not hard to get inside the computer and play with it.

"You can do a lot of this design with the Macintosh and FileVision," Brentano admitted. FileVision is a visual database which lets you design icons that can be moved around the screen. Each icon also represents detailed information, such as the focus, location and type of each light.

"But Apple has a certain snobbishness I object to," said Brentano, who believes that the Atari is the best 8-bit machine on the market. "Pong is the greatest thing since sliced bread."

LIGHTING BOARD

Brentano's dedication to the Atari has made it the computer of choice for a theatrical bulletin board he's starting.

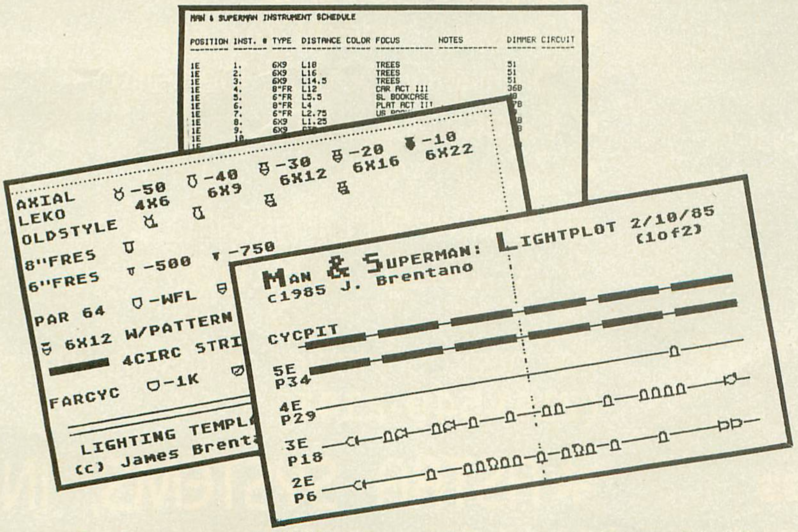
Based on an Atari 800, an MPP 1000C modem and two disk drives, Brentano's bulletin board will cater to the theatrical lighting community. It will have classified ads for jobs, equipment and so on. The board will also feature electronic mail, hints and tips for lighting designers, product reviews and comments.

"Eventually, I want to be able to dump a show's files to the bulletin board, so rental companies can log on, look at the file, and give me a price."

"A problem with theater is that we spend $10–20,000 per show. Any way to share information saves money," Brentano said. "There's a need for a clearinghouse of information."

Brentano's board will be the only Atari theater BBS in the country—and only the second theater board of any kind. It is called JCN (James' Computer Network). The phone number is (415) 562-3364.

"JCN—it's like HAL in '2001.' HAL was from one letter before IBM. JCN is one letter after."



The map and key above are used to tell theater electricians where to hang certain lights, what kinds of lights to use, and what focus each light requires.

ANTIC AT THE FIRST GEM SEMINAR
continued from page 12

difficult for professional programmers to work with.

Despite this, Wetmore was positive enough about GEM that he bought the Seminar disks and intends to invest over $4,500 for a souped-up IBM PC to run Lattice C. "But right now any GEM programs I publish will just be for the Atari," he said, "because it's the only computer that a developer won't be charged royalties for by DRI when the software comes out."

TURTLE PIANO
continued from page 10

As stated earlier, this program is really pretty simple. Readers are encouraged to make it more special. With a little tinkering, Turtle Piano could flash different colors with each note, extend for another octave or two, or include sharps and flats. The basic structure can accommodate all these changes and more. The only rule is to share your ideas by sending them to **Antic**.

Craig McBain is a third grade teacher in Mt. Clemens, Michigan. His wife wishes that Atari Logo had never been invented.

Listing on page 74

THE #1 SYSOP
continued from page 8

from the Japanese game Go, and SIG*Atari has an oriental sound," said Luks, pronouncing the name quickly and sibilantly, as one monosyllabic word.

MILLION-DOLLAR SIGS

SIGs are now the second or third largest moneymaker for CompuServe, bringing in millions of dollars a year, Luks said. And he has a well-defined niche in that enterprise.

Luks, the "Godfather of SIG*Atari", now works with a CompuServe sysop school and a test SIG where new SIG software is beta tested. He also helped launch the IBM SIG and works eight other boards besides SIG*Atari—as well as being OnLine Editor of the **Antic** CompuServe Edition.

Of course, all that takes a lot of time, "60 to 70 hours a week," Luks figured. As we've found out at **Antic** when we try to get in touch with Luks, the phone at his Greenwich Village apartment routinely gives off a busy signal till after 3 a.m.

And during regular working hours, Luks is a freelance computer consultant to a major Wall Street brokerage firm.

"Nowhere else can you find a more knowledgable group on the Atari world than on SIG*Atari," Luks said. SIG*Atari members gave constructive input to Synapse Software during its development of the Syn Series. More recently they've done the same for Batteries Included's HomePak.

"Our users decide what service they get and dictate policy," Luks said. However, Luks insists that users maintain a sense of decorum on SIG*Atari and that the board not become a haven for pirates.

FREE ACCESS

Luks is committed to bringing the SIG to as many Atari users as possible. If a large users group has never been in SIG*Atari before, Luks said he would try to arrange free access to the SIG for a weekend so the group could discover the magic for themselves.

"If we could get a new machine from Atari, we could have 50–60 public domain programs for the ST on the SIG*Atari," Luks claimed. Nor does it appear an idle boast. "On the Macintosh SIG, we had 50 programs when there were only five for sale in the stores."

"All they have to do is ship it. We'll do the rest," Luks promised.

Luks sees his demanding work with CompuServe as an investment in the future. "We're shaping the technology. Also I want to be able to live anywhere, to be geographically independent and able to work over the phone."



Let your joystick select program hues!

by JOHN FELTON

Take the tedium out of program color selection with this joystick-driven BASIC utility. Color Palette runs on all Atari computers of any memory size. Disk or cassette.

When the time comes to select colors for your new program, it's *hard* to choose among all the Atari's 128 colors and multiple luminances—if you can't conveniently compare similar shades. Until now, it was a tedious process to find the perfect green for your slime monsters, or just the right red for the sky over Barsoom. You might use SETCOLOR in a program, run it, look at it, break it, adjust it and start over again.

USING PALETTE

Color Palette gives you a better way to use the best home graphics computer on the market.

Type in Listing 1, check it with TYPO II, SAVE a copy, and RUN the program. You'll be looking at a joystick-driven utility for color selection. Listing 2 is the source code for the assembly language routines used in the BASIC program. You don't need to type it in.

The palette's screen is divided into two windows. In the top, you'll see 12 patches of color, each with its hue and luminance numbers and a color register value. The lower right corner is the background color.

The bottom half modifies the color of patches in the top half. Put the cursor over the patch you wish to modify and press the joystick button. You'll now be able to change colors by moving the stick up and down, and change luminance by moving the stick from side to side. Press the trigger to set that color and move on. To see your values, watch the X in the lower window.

That's all that there is to it. You might note that the top four patches are initialized to the Atari default colors. It is amazing how many programs use these four colors just because it's easier than looking for better ones. Well, no more.

A three-year veteran of the Atari, John Felton is a computer science major at Cal Poly in San Luis Obispo, California.

Listing on page 64

COLOR the COVER CONTEST



By JACK POWELL and MICHAEL CIRAOLO

When we invited submissions for our Color the Cover Contest, we didn't really expect a winner with the diligence, creativity and . . . uh . . . compulsiveness of **Paul Sedgewick**. His adaptation of the January 1985 **Antic** cover was artistically a match for any other entry—but it was also a mind-boggling technological tour de force.

The 23-year-old electrical engineer from Northridge, California spends his days producing delicate military microwave machines. And he put in over 100 hours re-creating **Antic's** cover with *redefined characters on two scrolling screens.*

Paul decided that the black and white **Antic** January cover needed at least five colors, if not more. He also didn't think one video screen was large enough.

Wanting the highest resolution possible, Paul chose ANTIC Mode 4, the only mode which offers both high resolution and five colors. (Mode 4 is a five-color text mode, counting the background.)

He then photocopied the cover, enlarged it and graphed it on a grid approximately 40 squares wide and 50 down—each block representing one mode 4 character.

Next he graphed the cover in even finer detail, dividing each block into a 4×8 pixel matrix.

34 CHARACTER SETS

Having done all this preliminary paperwork, Paul finally reached for his own trusty character set editor. Using the editor, Paul redefined each character block to match the cover. However, there are only 256 characters in a character set—just enough to draw three screen rows.

Rather than re-use characters for different portions of the picture, Paul used display list interrupts (DLI) to actually change character sets while the screen was being drawn. After each three rows of characters on screen, a different set was pointed to in memory.

Paul decided that five colors really weren't sufficient, so he used a vertical blank interrupt (VBI) to draw the picture twice, thus creating extra colors by superimposing the hues. Two overlapping hues of the same luminence created a third color; two hues of different luminences created a vibrating, flickering effect.

The VBI was also used for vertical fine scrolling between two screens, which meant the display list instructions had to be continually refigured.

What began as a simple Color the Cover Contest entry ended in a four-screen phantasmagoria. The VBI alternates between two screens, and the vertical scroll switches between two more. After 100 hours of work, Paul had created the necessary 34 complete character sets. With an accompanying assembly language program, the entire binary load file occupied 316 disk sectors!

. . . AND RUNNERS-UP

Nearly 150 readers mailed in entries to the contest. Just about half the entries used the KoalaPad or Atari Touch Tablet with accompanying Micro Illustrator (AtariArtist) software. Interestingly the second most used computer graphics tool was "Price's Picture Painter" the powerful program from **Antic's** September, 1984 issue.

Tim O'Connell, of Mastic Beach, New York used a KoalaPad to produce his strong head-and-shoulders close-up of Utility Man. Our distinguished panel of judges (**Antic** editorial and art departments) particularly liked the way this U-Man seemed to be posing for a hero portrait.

Alfred Gomez of Las Cruces, New Mexico spent several sleepless nights to produce his entry. Gomez demonstrated his good taste by depicting a miniature **Antic** magazine hanging from Utility Man's belt, although this is too small to see in our screen shot. Our judges were also pleased with the excellent use of contrast and color. His successful ef-

and the winner is . . .

Paul Sedgewick



forts "burned my rendition into the retinas of my eyes." He was the only entrant to use the Fun with Art cartridge.

Montreal's **Gaston Aladin** used the Atari Touch Tablet and his own software to produce his highly impressionistic entry. Aladin's work demonstrates interesting, complex background textures captured in a fluid sense of movement.

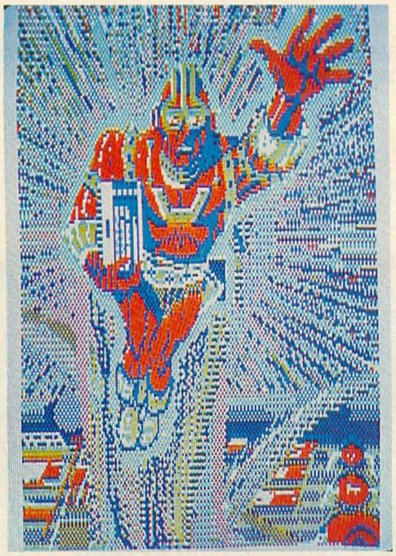
Nick Turner was "inspired to go to the limit" of his patience in recreating our cover. This programmer from Mountain View, California used AtariArtist cartridge with Atari Touch Tablet to produce a Utility Man that closely resembled our cover. Turner displayed well-balanced colors, a good sense of proportions, and well defined shapes.

When she read about the Color the Cover Contest, **Marta Taylor** ran out and bought a KoalaPad to produce her entry, one of the more abstract renditions we saw. Taylor, who hails from Douglasville, Georgia, depicted Utility Man surrounded by a fanciful collection of balloons. While not a true copy of the cover, the entry contains unique imagination and a charming, simplistic innocence.

continued on page 24

runners-up . . .

Gaston Aladin



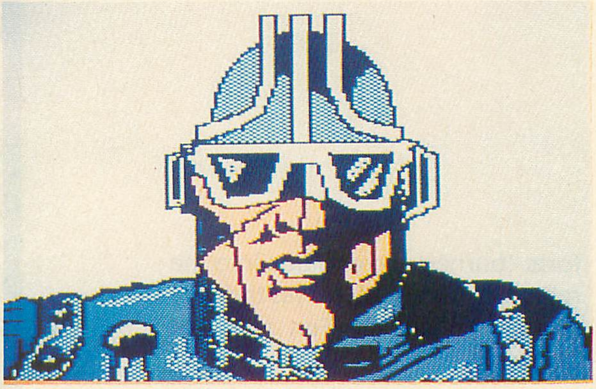
Runner-up winners are not presented in any particular order on these pages. Color The Cover Contest first prize is an Indus GT disk drive. Runner-ups receive their choice of any single item in the **Antic** Arcade Catalog.

Antic Disk subscribers: You will find the winning contest entry as a bonus on your August disk NEXT month. It was too large to fit anywhere on the current issue's two-sided disk.

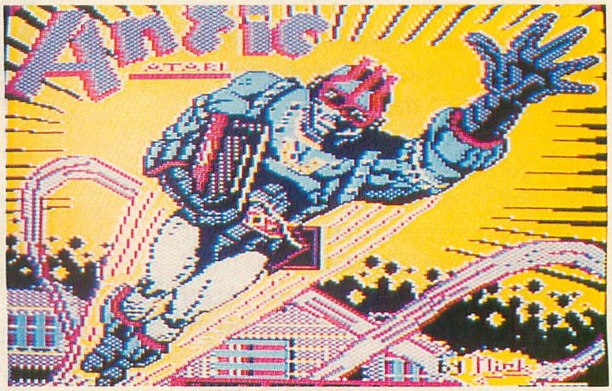


Alfred Gomez

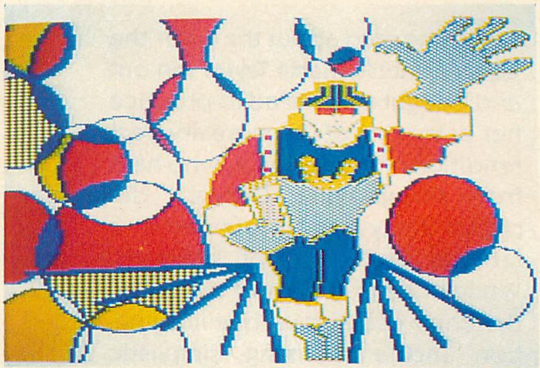
Tim O'Connell



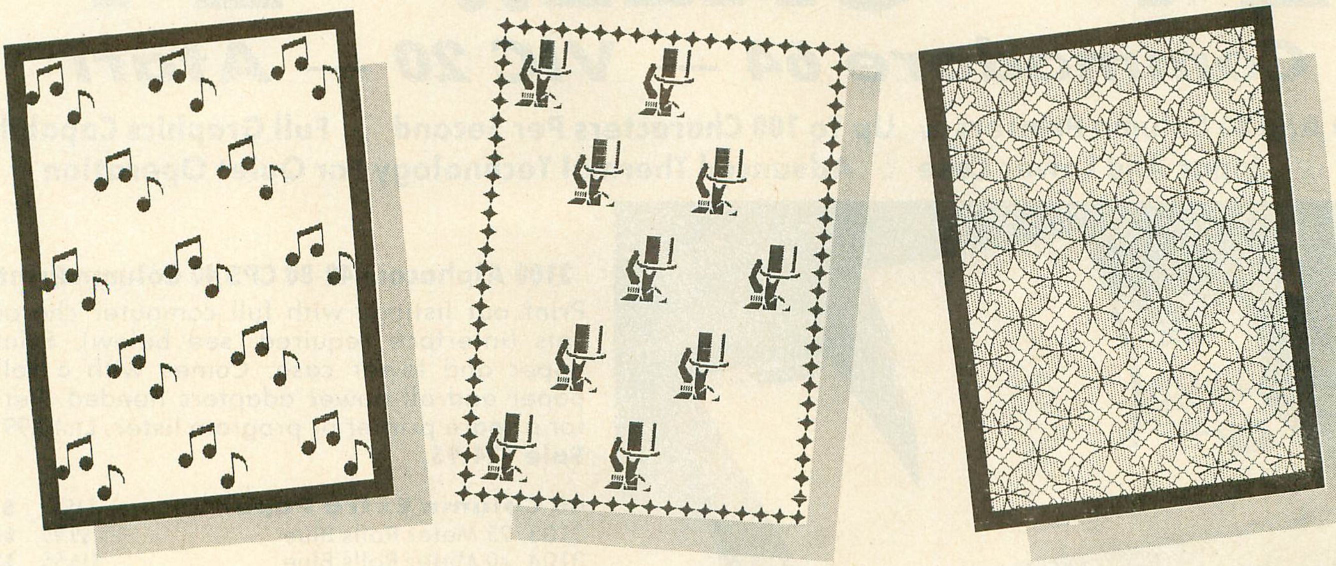
Nick Turner



Marta Taylor



PRINT SHOP



WELL WORTH THE WAIT!

by JACK POWELL, **Antic** Technical Editor

Yup. Looks like the dry spell for Atari software has finally ended. At last, the big guns are coming out: EPYX is marketing the fabulous Lucasfilm games, PaperClip is on the way, and Alternate Realities will finally become a reality—courtesy of Datasoft.

Then just the other day a bright yellow box came in from Broderbund. The Atari edition of **Print Shop** had finally arrived!

Print Shop is a graphics printing program that Apple owners have been enjoying for quite a while. It's currently Broderbund's hottest selling item, and the reason why is easy to see. From the bright yellow packaging to the foolproof programming, Print Shop is so darn friendly you feel like Bambi has just toddled into your Atari.

When the package arrived, we tore off the wrapping and booted the disk. The very first page of the reference manual said to ignore the documentation, start up the program and just follow the prompts. Within minutes, the **Antic** offices were littered with gaily decorated printer paper.

If you've always dreamed of being an art director, Print Shop is exactly what you've been waiting for. You'll be cranking out flashy greeting cards, stationery, banners and signs, featuring clever graphics and a variety of different type-styles. You can decorate with ready-made graphics icons that are supplied, or design your own. (My own "Opus The Penguin" has been very well received.) The results are remarkably professional.

The program is entirely menu-driven and leads the user easily through the options. In the tradition of all Broderbund's Apple-oriented programs, there is an almost complete lack of sound, but this is essentially a utility so it's not too bothersome.

Before you rush out and buy Print Shop, make sure you have a dot-matrix printer capable of graphics. No matter how nice the program, it won't work on your letter-quality Atari 1027 printer. Print Shop handles most graphics printers. They're all listed on the box. If in doubt, call Broderbund.

As fun as it is, Print Shop has its limitations. You've eight type-fonts to choose from and there's no way to design your own. The disk comes with 60 icons or graphic designs. And you may also design your own using the Print Shop graphics editor, but as usual with multiple utility packages, the graphics editor is a crude joystick/Koala Pad pixel editor which does not allow for patterned fills or other desirable sophistications.

It would certainly be nice if we could use one of the many versions of Micro Illustrator to design our icons, but this is not the case. And, since the disk files have been hidden,

continued on page 28

ATARI®

Lowest Price In The USA!

Computer System Sale

• Students • Word Processing • Home • Business



LOOK AT ALL YOU GET FOR ONLY **$449** SYSTEM PRICE

	LIST PRICE	INDIVIDUAL SALE PRICE
① Atari 800XL 64K Computer	$299.00	$109.00
② Atari 1050 127K Disk Drive	399.00	189.00
③ Atari 1027 Letter Quality 20 CPS Printer	399.00	199.00
Atari Writer Word Processer	79.95	39.95
Atari BASIC Tutorial Manual	16.95	12.95
TOTALS	$1193.90	$549.90

All connecting cables & T.V. interface included.
☆ Monitors sold separetly.

SAVE $100
All 5 ONLY
$449.00
SYSTEM SALE PRICE

MONITOR OPTIONS:

	List	Sale
☆ 12" Hi Resolution Green or Amber Screen	$199.00	99.00
☆ 13" Hi Resolution Color	$399.00	195.00

Add $9.95 for Connection Cables and $10 for UPS

15 DAY FREE TRIAL. We give you 15 days to try out this ATARI COMPUTER SYSTEM!! If it doesn't meet your expectations, just send it back to us prepaid and we will refund your purchase price!!
90 DAY IMMEDIATE REPLACEMENT WARRANTY. If any of the ATARI COMPUTER SYSTEM equipment or programs fail due to faulty workmanship or material within 90 days of purchase we will replace it IMMEDIATELY with no service charge!!

Best Prices • Over 1000 Programs and 500 Accessories Available • Best Service • One Day Express Mail • Programming Knowledge • Technical Support

Add $25.00 for shipping and handling!! Dept. ANT/6
Enclose Cashiers Check, Money Order or Personal Check. Allow 14 days for delivery. 2 to 7 days for phone orders. 1 day express mail! We accept Visa and MasterCard. We ship C.O.D. to continental U.S. addresses only.

COMPUTER DIRECT

We Love Our Customers
Box 1001, Barrington, Ill. 60010
312/382-5050 to order

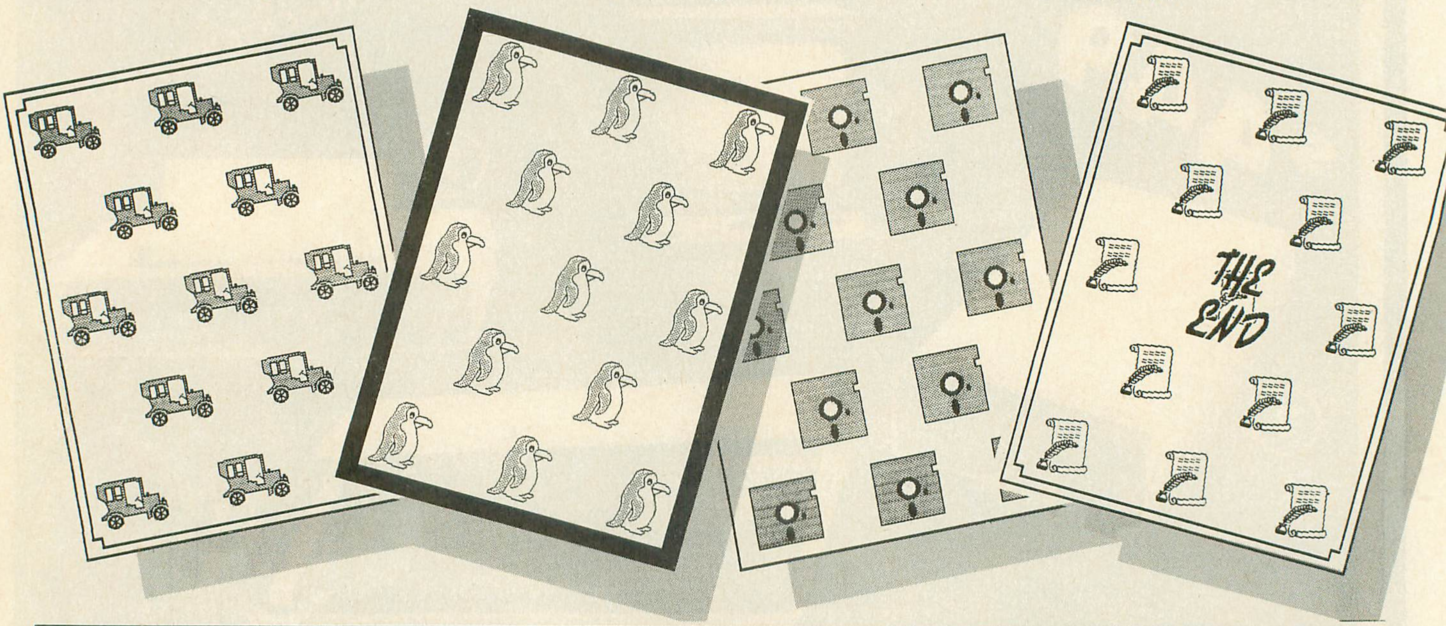
PRINT SHOP
continued from Page 25

a programming hobbyist would have a difficult time writing a conversion program.

One frustrating omission is Print Shop's inability to make labels. What about all my home-made peach preserves and canned prunes?

Nevertheless, Print Shop is the kind of program that should appeal to a huge audience. I can't think of a better computer gift for children—if the adults will let them get their hands on it. We really should all thank Broderbund for converting this hit software to the Atari. Now if they would only buckle down and get to work on an Atari conversion of Championship Lode Runner. . .

PRINT SHOP
Broderbund Software
17 Paul Drive
San Rafael, CA 94903
(415) 479-1170
$44.95—48K disk



Color Inkle Loom

Your Atari master weaver

by GERALD M. HAGOPIAN

Design intricate 39-thread weaving patterns for the inkle loom with this BASIC program. (If you're a weaver, inkle loom won't sound strange.) Color Inkle Loom works on all Atari computers with 24K memory.
24K cassette, 32K disk

Serious weavers regularly face the process of selecting colors and patterns for their various looms. And as you know, displaying intricate color patterns is a job that's duck soup for the Atari. With the Color Inkle Loom program, you can design patterns for use with the inkle loom which—in weaver's jargon—is a two-harness plain weave loom.

This program should be a good start for **Antic** readers who might wish to develop design software for looms that are more complicated than the widely used inkle.

ATARI WEAVING

Type in Listing 1 and check it with TYPO II. SAVE it before you RUN it.

After the opening screen, you'll see a menu. You can choose to Create a new weaving, Save a weaving, Load a previous weaving, or Exit the program.

If you are creating a new weaving, you'll be asked for the number of threads per harness, up to 39 each. After typing a number, you'll be asked if you're sure. Type [Y] to continue, anything else to go back.

Then choose colors, starting with harness #1, thread #1, followed by harness #1, thread #2, and so on. In each case, you'll be asked for the color letter—which you'll take from the marked band of colors across the screen. Once again, you'll be asked if you're sure. Type [Y] to continue, anything else to go back.

When you're done choosing colors, you'll be asked if you're ready to weave. Type [Y] to continue. You can start weaving, or change the intensity of the colors you've chosen. With the spectrum band's range of 16 colors, and the Change Color Value option, you can produce any of the Atari's colors.

Gerald M. Hagopian is a freelance consulting designer working with consumer products and interior design.

Listing on page 60

Play it Again, Atari!

They laughed when I sat down at the 800 XL...

by NAT FRIEDLAND, **Antic** Editor

Okay, I confess. Before I settled on writing I was a committed would-be musician. As a kid and teenager I must have spent hundreds of hours teaching myself piano, rhythm guitar and chromatic harmonica. I played first trombone in the high school band, thus becoming automatically eligible for a music scholarship to state teachers college if I had wished to go.

It was relatively easy for me to pick up the rudiments of playing different instruments. But there would always come a time—much too soon—when I ran into the upper limits of my musical coordination. I could never seem to really master any instrument.

I used to daydream about some kind of future electronic musical instrument coming along that would automate things the things I couldn't get my fingers to do and let me express my musical ideas without being an instrumental whiz. Little did I know those instruments would become a reality—even an affordable reality—in less than two decades.

Recently my musical frustrations have been much more localized. It seemed as if all the good musical software and plug-in computer piano-keys from companies like Sequential Circuits or Passport were being made only for the Apple II or the Commodore 64.

1985 ATARI MUSIC

However, in 1985 all this has changed. Some truly phenomenal new musical peripherals for our favorite personal computer are about to make the Atari the new champion of computer-assisted musicmaking.

In this article you will read about:

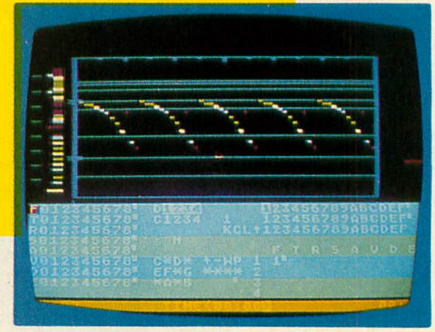
- A new kind of "music generator" software that lets you compose and improvise in real-time at the Atari keyboard. The four-voice sound is tracked by colorful geometric graphics. Press a couple of Atari keys and you'll feel like the next Brian Eno.

- A software and "black-box" product that lets your Atari emulate an advanced 16-track digital recording studio.

- A remarkably powerful new MIDI synthesizer that sells for no more than what an Atari disk drive used to cost.

1. VIRTUOSO

At a music studio in Queens, New York last year you'd find three kids at a time sitting in front of Atari computers and listening on earphones while geometric patterns of color flashed across the video screen. These

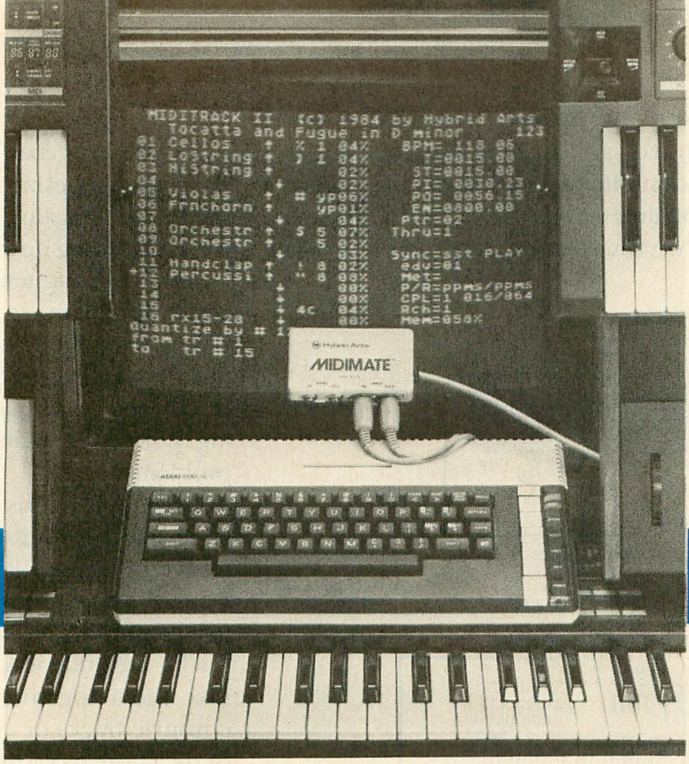


VIRTUOSO screen.

kids were taking two-hour lessons in creating music on the **Virtuoso** sound generator and when the sessions were over their parents often had to drag them away from the machines.

Virtuoso is such a unique new approach to musicmaking that it's not easy to describe. It's one of the closest things in the real world to the multi-arts competitions that Herman Hesse wrote about in his classic literary fantasy *The Bead Game*. In that book, Hesse wrote about chess-like contests where one player's move might be a theme from a symphony and the opponent's countermove could be a line of a poem or a section of a painting . . .

Virtuoso gives you a user-friendly method of tapping the extremely fast and powerful changes that a computer can control in every aspect of music performance. It bypasses the limits of traditional musical notation



MIDITRACK II

and uses an almost self-explanatory color graphic display that delivers mathematical insights into the structure of music.

USING VIRTUOSO

You'd enter a musical pattern into Virtuoso from the Atari keyboard, or call up one from about 480 that could be stored on a single disk. The pattern would start sounding and the lines of colors would trace it visually. At this point you could start creating all sorts of changes in the pattern—which you would hear and see *immediately*.

As the pattern was playing, you could change its speed, rhythm, pitch, tone, volume, key scale, etc. You could enter new patterns any time. There's even a Future mode where you can enter changes before they are due to be played. The effect of controlling so much musical power so effortlessly feels something like conducting an orchestra at the same time as you are composing the music that it plays.

In technical terms, Virtuoso is a sound generator that produces four voices from the POKEY chip. You can make instant real-time changes in the voices in any of six parameters. Four computers running Virtuoso can be linked together to have up to 16 independent channels controlled by one Atari.

As a sound editor, Virtuoso can synchronize multiple voices with 1/60 of a second accuracy and tune them within 10 steps of intonation. Any musical passage can be moved anywhere, saved, and replayed in any key and in virtually any rhythm.

COMING SOON

This groundbreaking product is a collaboration between former Julliard Music Professor Joseph Lyons and Frank Schwartz, a highly experienced programmer and electronics designer. Originally, Virtuoso was financed by Warner Leisure Software, who naturally wanted it for the Atari and in cartridge form.

After Warner Software shut its doors last year, Schwartz and Lyons obtained new funding and are hoping to have Virtuoso on the market by August. At this point, Virtuoso is to be on disk, available for either the Atari or Commodore 64, and priced at about $50.

Not only that, a $150 MIDI interface for Virtuoso is also being readied for August release. Virtuoso will therefore be usable as a visual language for MIDI controllers—not only for music, but also for lighting and sound effects, lasers, etc. Once again, shades of Hesse's *The Bead Game*.

Lyons and Schwartz are as enthusiastic about the Casio CZ-101 synthesizer as **Antic** is, and Virtuoso will definitely run on this outstanding electronic instrument—which will provide even greater power, versatility, sound quality and handling ease than the Atari POKEY chip.

PLEASE NOTE that Virtuoso is a product that is still under development and has not yet been released at this writing. **Antic** will print more news of Virtuoso as soon as it becomes available, so please do not phone or write us asking where to get it yet.

How does **Antic** know that Virtuoso is for real? There are two reasons. 1. We have heard (and seen) Joe Lyons play four-part Bach Fugues on it. 2. **Antic** has a first-generation Virtuoso cartridge that Frank Schwartz gave us.

Our prototype Virtuoso cartridge is packed solid with microchips and actually a plug-in board. Its music generating functions are 100% in working order, but figuring out how to play it from only the skimpy documentation notes is not too easy. At present you'd need Lyons standing over your shoulder to explain things, the way he does in his studio lessons.

Thats why the final development work is concentrated on making Virtuoso even friendlier to operate. There will be icon menus, an inexpensive membrane keyboard for musical input (if you're not using a MIDI instrument), and six levels of complexity that will gradually take you from beginner to expert status.

2. MIDITRACK II

MIDITRACK II has been wowing them at computer shows and musical instrument shows since last fall. It's available at various professional-music stores around the country or by mail from the manufacturer for $349. (Detailed manufacturer information will be found at the end of the article.)

Interestingly, your Atari will be the least expensive component of this

continued on next page

music system. Bob Moore of Hybrid Arts, makers of MIDITRACK II, gives a slightly surprising reason why the Atari was chosen to drive the system. "The Atari is the sturdiest of the inexpensive lightweight computers," he said. "We believed it would have the best chance to survive a long professional road tour."

MIDITRACK II disk software and the included MIDIMATE interface box work with any Atari that has 48K memory. The Atari itself does not produce any sounds with its POKEY chip here. It simply acts as the controller for up to 16 channels of information transmitted by MIDI instruments.

WHAT'S MIDI

MIDI stands for Musical Instrument Digital Interface. It's a set of electronic standards—just as ASCII, RS-232 and Parallel Centronics are standards—that allows electronic musical instruments to coordinate and exchange digitally encoded sound information.

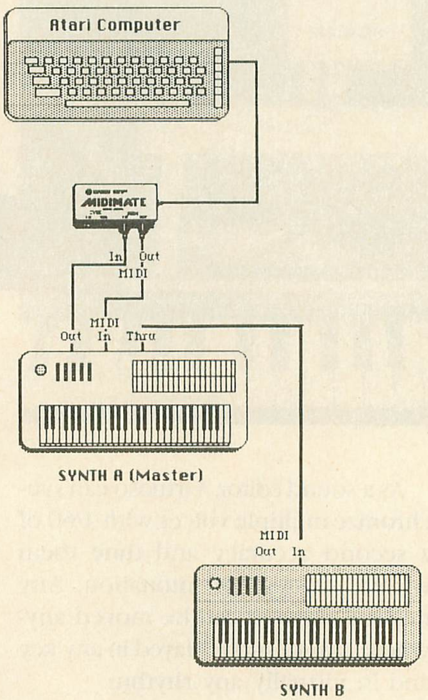
Moore, who was primarily a Hollywood studio musician before coming up with the idea for MIDITRACK II, said, "If you've already got an Atari and disk drive, it should cost you no more than $3,000 to have a fully professional digital recording setup. (To just have fun with your Atari music system, you could get away with $500 or less. More on this later.)

What you need for a fully professional system is a main synthesizer, a drum machine, and probably a second synthesizer to give you a bit more variety of sounds. The second synthesizer doesn't even need to have a keyboard because you can play it from the main synthesizer.

At the other extreme, you could theoretically daisychain huge gangs of MIDIMATEs and electronic instruments. You could mix 16 completed tracks onto a single track, make 15 new tracks and mix everything down to track 2, and then repeat the process. You could run a symphony orchestra of synthesizers from a single Atari, even a stadium filled with synthesizers. . .

Normally the way you'd operate a MIDITRACK II system is something like this: First you'd set up a drum pattern and record it on track 1. Next you'd adjust your synthesizer to sound like a bass and play an accompaniment onto track 2. With your "rhythm section" in place you could

Figure 1



A typical system hook up is shown in figure 1. Note that synth A is the master synth because both its MIDI IN and MIDI OUT are connected to the MIDIMATE interface box.

then start layering all sorts of interesting synthesized sounds on top to make melodies and harmonies in the rest of the available tracks.

STUDIO IN A BOX

Once you were finished, you would have a fully edited arrangement for MIDI instruments which you could then record on tape for combination with vocals or non-MIDI instruments. The length of the music you could save would depend somewhat on how many notes were in the piece. The limit per file is 3,000 sequenced notes.

By the way, usually you can simultaneously call up more than one track from a single MIDI instrument. Many synthesizers could give you as much as 8 simultaneous tracks.

On the whole, the MIDITRACK II documentation is excellent. Once you have plugged everything in, the manual suggests that you simply press your Atari spacebar, play something on your synthesizer, and then press the spacebar again. That's all it takes for a recording and playback!

One of our testers kept losing his music at first, every time he tried to save a track. But once he figured out that this was caused by holding down the Inverse Video key too long during the save command, there were no problems.

MIDITRACK II is designed to operate like a professional multitrack tape recorder. So it contains all the features you would normally expect to find in a recording studio. All 16 tracks are independent unless you mix them together. You can synchronize tracks or change the speed of the entire recording. You can overdub or transpose tracks. You can automatically locate any spot on the recording. You can "punch in" anywhere to record difficult passages one note at at time.

MIDITRACK II even supports the advanced technique of quantization, or autocorrect. For example, if your timing was a bit uneven when you were trying to play that flashy bass part you could set the notes to automatically come out on the beat.

3. CASIO CZ-101

Most Atari owners who buy MIDITRACK II will probably decide to use the new Casio CZ-101 synthesizer as their primary keyboard. That's because the CZ-101 sells for about one-fourth the price of any comparable synthesizer! It lists for $499 but has been on sale at Macy's for as low as $300.

The instruments that Bob Moore brought along to demonstrate MIDITRACK II were the Yamaha DX7 synthesizer which has a list price of $1,995 and the Yamaha RXII drum machine which lists for $895. Both of these instruments are very popular with professional musicians and are not considered unusually high-priced in comparison to the competition.

At the time, Moore told **Antic** that

a new low-priced but powerful synthesizer from Casio was due to be released shortly and it would apparently be at least somewhat comparable with the DX7. Well, the CZ-101 was shown at the Consumer Electronics Show and Casio kindly let us have one to use with our MIDI-TRACK II.

ST OF SYNTHS

We swiftly discovered that the CZ-101 is considerably more than merely a stripped-down version of the DX7. In fact, this Casio could almost be considered the Atari ST of sythesizers—it delivers far more "power without the price" than anything else in its class. Despite the Casio brandname we are talking about a real synthesizer here, not an "electronic music-maker" with one-key chords and preset drum-bass patterns. (It doesn't have a built-in speaker either.)

Figure 2

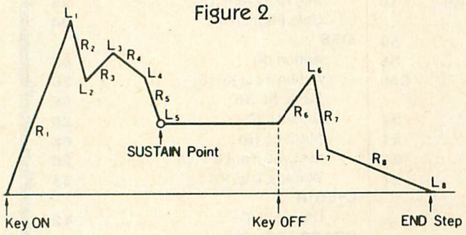
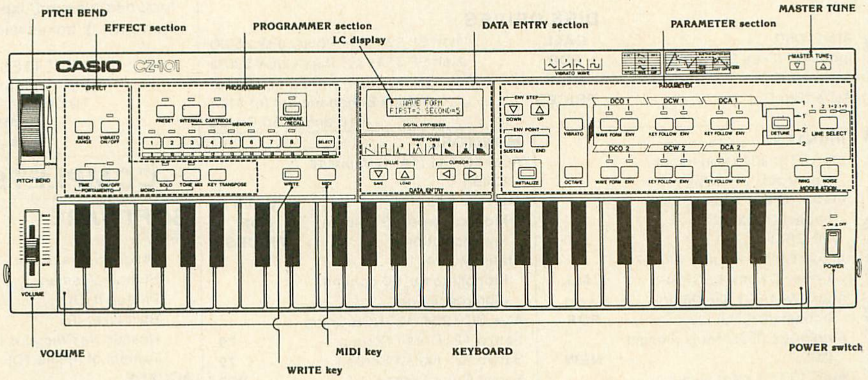


Fig. 2 shows an example of an envelope using all 8 steps. In this example, there are two attacks before the Sustain Point as well as a third attack after releasing the keys. This shows how you even have the possibility of setting an "after-envelope".

In many ways the CZ-101 is even *more* versatile than earlier, more costly synthesizers. A review in the March, 1985 issue of "Keyboard," the top magazine for electronic keyboard players, concludes, "The CZ-101 makes good use of the latest digital technology. Its attractive features include seven excellent envelope generators, good-sounding waveforms, and several doubling modes for building up complex timbres. As an inexpensive and versatile MIDI slave module, it could be a very effective addition to almost any stack."

Upon translation from synthesizer jargon, what this means is that the CZ-101's strongest point is its wide-ranging capability of creating and manipulating synthesized sounds. It has *more* waveforms, envelopes, oscillators and *more* ways to combine these soundmaking elements than most previous synthesizers.

Figure 3



CASIO CZ-101

In this instrument you'll find a full assortment of standard high-end synthesizer features such as pitch-bend wheel, ring modulator, portamento, octave shift, detune control, phase distortion sound generator.

16 INSTRUMENTS

The CZ-101 starts you off with 32 factory-preset sounds—flute, electric piano, violins, organ, etc.—that range from okay to pretty good. You can reprogram 16 of these sound "patches" to hold your own sound creations (you can bring back the factory patches anytime). Also there's a slot for additional 16-patch programmable cartridges.

People who play piano by ear and can only play in one key (usually either all white notes or all black notes) will deeply appreciate the transpose button that will instantly shift you into even the most complex key (four flats, five sharps, etc.).

The CZ-101 has 49 keys of standard "mini-keyboard" size. Purists may insist that only full-size keys will do, but personally I enjoy the feeling of spanning left-hand tenths as effortlessly as I would reach octaves on a full-size keyboard.

A GREAT TEAM

The CZ-101 works in combination with MIDITRACK II remarkably effectively. You don't need to be a musical genius to record and playback flashy multitrack compositions featuring your own synthesized sounds almost as soon as you've got your system cabled together.

And you can dramatically change the synthesizer voicings during playback and hear your new sounds in real time. Or if you tinker with the playback of the demonstration songs provided with MIDITRACK II you can try out sounds as unique as a Mozart Sonata being played on a vibraphone or jazz organ.

So tune up your Atari and unlock your creativity. With MIDITRACK II, the Casio CZ-101 and Virtuoso, you might very well be world's next musical genius!

MANUFACTURERS

MIDITRACK II
Hybrid Arts
P.O. Box 480845
Los Angeles, CA 90048
(818) 508-7443
$349—48K disk

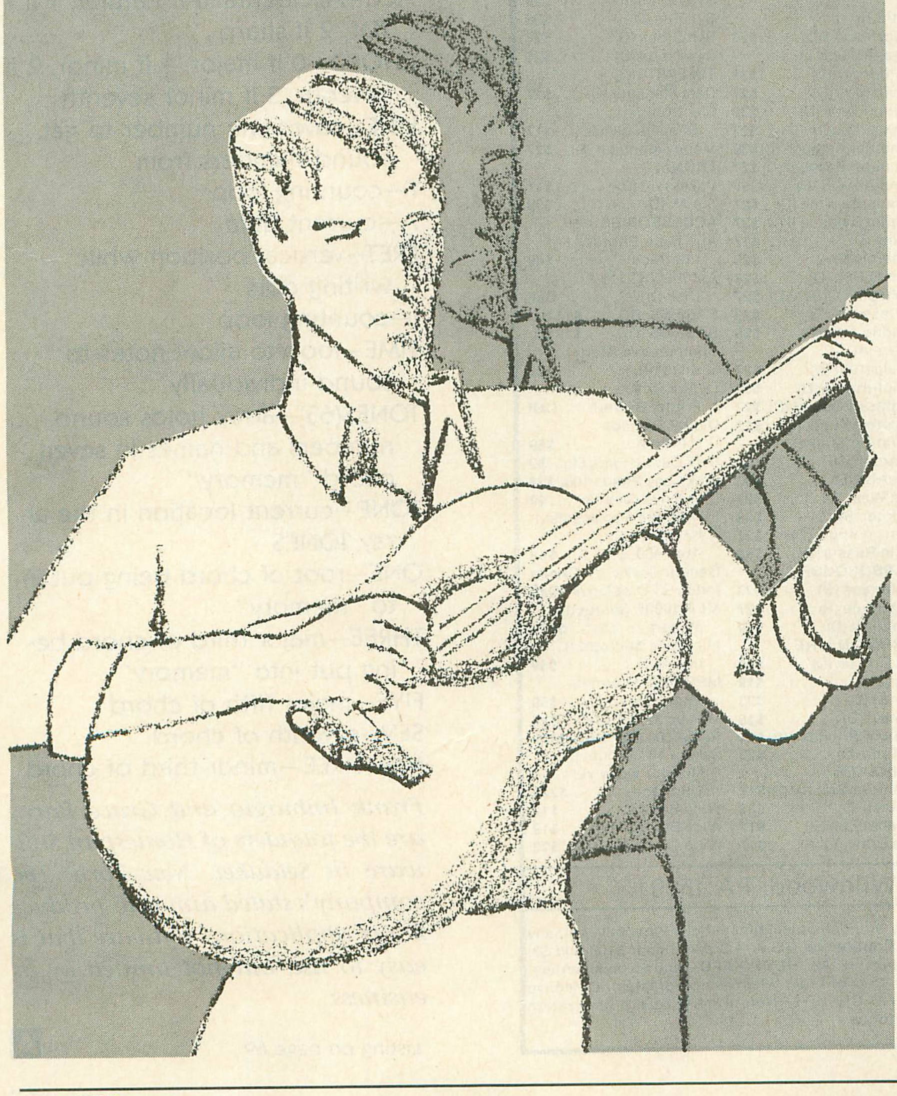
CZ-101 SYNTHESIZER
Casio, Inc.
15 Gardner Road
Fairfield, NJ 07006
(201) 575-7400
$499 (Suggested list)

VIRTUOSO
Enhanced Technology Associates
125 W. Duke Ellington Blvd.
New York, NY 10025
$50—48K disk
$150—MIDI interface
(Available August 1985 or later)

GUITAR TUTOR

Learn and play guitar chords on your Atari

by FRANK IMBURGIO and GRACE BARRY



A program that diagrams and plays simulated guitar chords. The BASIC listing works on all Atari computers of any memory configuration.

My great new Program In The Works wasn't working at all. Rather than break my new 800XL, (which was a serious consideration at this point), I reached for my guitar. I played a song or two while staring at the screen, and the idea for Guitar Tutor was born. Why not create a program to generate, diagram, and play guitar chords: major, minor, seventh and minor seventh chord combinations?

To use the program, first type in the BASIC listing. Check it with TYPO, and save an extra copy as a backup.

A music background is not necessary to understand or play with "Tutor"! The first thing you will see is "Pick a note to build a chord on." Take your pick from the list given, A through G. Then you will be asked to pick a natural, flat, or sharp. After you do that, you can pick whether you want your chord to be a major, minor, seventh, or minor seventh.

Now the screen will diagram the chord for you! These are standard guitar chord diagrams used in music books. Vertical lines represent the six strings of the guitar. Horizontal strings represent the frets. Dots represent where you press your fingers. X's represent strings that are not strummed, and O's represent strings that are strummed "open", with no finger pressing on the string. If two dots are on the same fret, "bar" with your finger—press more than one string down with the flat of your finger.

If you press "S" at this point, you will hear the four individual tones which make up the chord you have chosen. To the right of the diagram you will see a vertical line of eight numbers, the eighth reading REST. By pressing a number from one through seven, you can put the chord on the screen into a "memory." Now press [P] and you can play your chords by

continued on next page

pressing the numbers you have assigned them. Play chords in any order and of any duration you want! Press [8] and you get no sound.

So as you see, you can also use Guitar Tutor to play a simple chord accompaniment if you feel like taking a break to sing a song while you're programming.

GUITAR TUTOR VARIABLE LIST

TOP$
BOT$
STRING$
FRET$ Graphic patterns to draw box
PATTERN(20)—Array holds DATA line numbers and is later used for sound numbers
K—value returned from keyboard
X—GOSUB 600 returns with X as a DATA line number
TONIC—adjusted note (i.e., B# reads data from C natural)
ACC—Accidental:0 if natural, 1 if flat, 2 if sharp
CHORD—0 if major, 1 if minor, 2 if seventh, 3 if minor seventh
LINE—saves line number to get sound numbers from
F—counting loop
A—current data
FRET—vertical position while writing dots
I—counting loop
TIME—loop to allow notes to sound individually
TONES(63)—array holds sound numbers and names in seven chord "memory"
TONE—current location in the array TONES
ONE—root of chord being put into "memory"
THREE—major third of chord being put into "memory"
FIVE—major fifth of chord
SEV—seventh of chord
MINTHREE—minor third of chord

Frank Imburgio and Grace Barry are the founders of Homespun Software in Setauket, New York. The company's stated aim is to produce home applications software that is easy to use, but not limited by its easiness.

Listing on page 69

the Musician

Type-in "music construction" software

by ANGELO GIAMBRA

Powerful and versatile "music construction" program. You can easily compose songs with three-part chord backgrounds—or simply copy the notes and chord symbols from sheet music. This BASIC program runs on any Atari computer with 32K memory. Disk or cassette.

With The Musician, you can easily enter and hear playback of any song on your Atari—with a full background of three-part chords. Only an elementary knowledge of music is required. In fact, you can simply copy your favorite songs directly from sheet music. You set The Musician to play harmony chords by simply naming them—the program fills in the required notes!

Type in Listing 1, checking it with TYPO II and SAVE it to either cassette or disk. Antic Disk subscribers will find a short demonstration song bonus file. Here's how to use The Musician.

USING MUSICIAN

When you RUN The Musician, you will see an introductory screen for a few seconds as the program initializes. A music staff will then appear on the screen with a single red note.

Use your joystick to move the note up and down on the staff. If you are working from sheet music, simply place the note in the same position you see it on the sheet music. To enter the note, press the joystick button. The note will turn white. Then The Musician will briefly play your chosen note and a new red note will appear.

TIMED NOTES

Placing your note on the staff just gives you the basic pitch. You must still set time values, indicate if a note is sharp or flat, and enter rests or dotted and sustained notes.

You select these various note types by pressing keys on the Atari. Here is how it works.

Whole notes	W
Half notes	H
Quarter notes	Q
Eighth notes	8
Sixteenth notes	6
Thirty-second notes	3
Dotted notes	.
Sustained notes	S
Sharp notes	#
Flatted notes	B
Natural notes	N
Rests	R

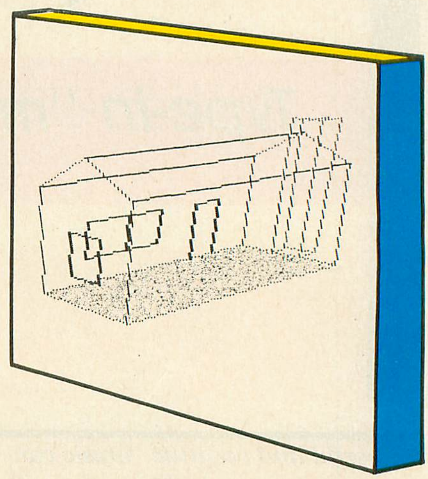
A little music information here: There are various "time signatures" possible at the start of a piece of music. In the most common types, 4/4 (standard) and 3/4 (waltz), a quarter note would be held for one beat, half note is two beats and whole note is four beats. An eighth note is half a beat, sixteenth note is one-fourth of a beat and a thirty-second note is one-eighth of a beat.

Dotted notes increase the time a note is held by half its normal value.

continued on page 50

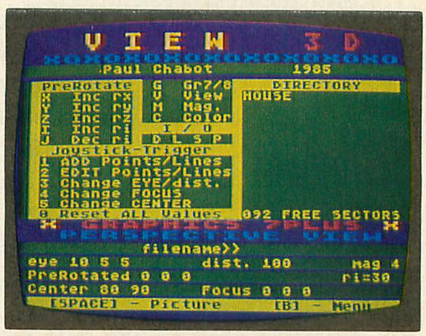
VIEW 3-D





Rotate and zoom 3-D images in ACTION!

by PAUL CHABOT



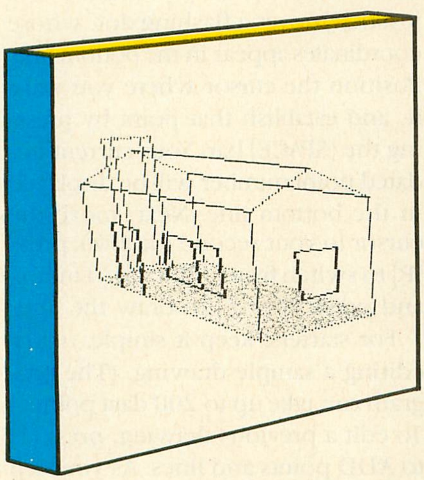
Create 3-D wire-frame outline pictures in your Atari's highest resolutions, Graphics 8 and Graphics 7+. Magnify, shrink, rotate, and otherwise shift your view of the 3-D picture easily and fairly quickly. Requires ACTION! cartridge, disk drive and 48K memory. Antic disk subscribers can run VIEW3D.EXE without the ACTION! cartridge. Disable BASIC and use the L option from DOS 2.0S. Disk or cassette.

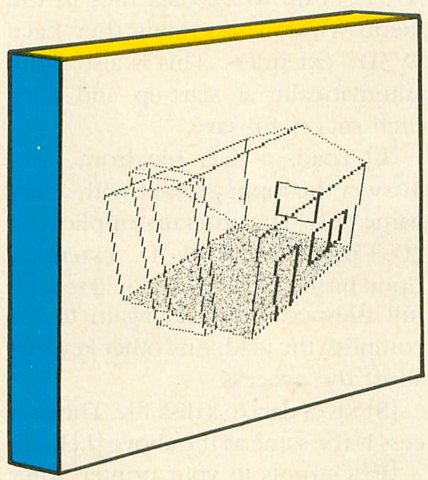
*When Paul submitted View 3-D to **Antic**, we saw it was easily the largest ACTION! program any magazine had considered publishing. But in recent months, we have received so many letters from readers wanting ACTION! that we thought it was time for a monster example of programming in this powerful Atari language.*

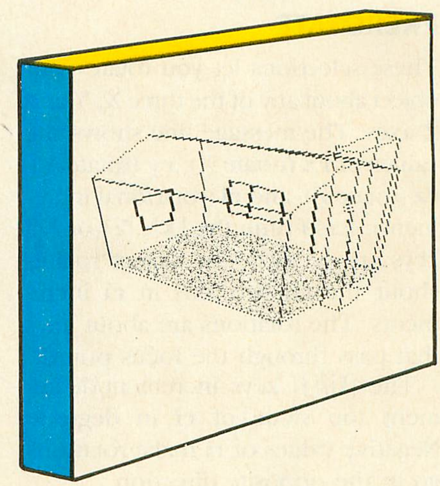
Be warned: there are ten separate program listings, nine of which are dependent on and INCLUDEd into the tenth to form one main program. Because of the nature of ACTION! there is no TYPO II, so type patiently and carefully. The results will be well worth it. —ANTIC ED

There are different approaches to 3-D viewing. You can leave the viewing point ("eye") fixed and rotate the object. Or you can think of the object as fixed and change the location of the eye. These are mathematically equivalent, but conceptually quite different to most people.

Also, should the projection be perspective or orthogonal? Where should the focus be placed? View 3-D will allow any combination of these variations and more. To manipulate a 3-D frame quickly, you need faster number crunching than BASIC provides. The answer is ACTION!, the cartridge-based programming language from Optimized Systems Software, which is becoming increasingly popular with serious Atari programmers.







TYPING IT IN

View 3-D is one program, but it has been split into ten files. Listing 10, called VIEW3D, is the main file which INCLUDEs the other nine. If you look at the beginning of listing 10, you can see the name of the other files.

Type each file in the order they are INCLUDEd in Listing 10. Each subsequent file shares procedures from previous ones, none may be compiled or run independently. You can partially check your work by compiling programs accumulatively in the order in which you type them. For example, GR78M may be compiled alone. After typing in MISC1, create a temporary third program which INCLUDEs GR78M and MISC1. This third program, when compiled, will compile the first two, and so on.

VIEW3D is too large to be compiled and run from the ACTION! editor. When all your files are properly typed in, clear the editor and, from the monitor, type: C "VIEW3D.ACT". After the compilation is complete, type [R] and away you go.

THE PROGRAM

The first thing you should see is the menu screen. View 3-D alternates between two screens—the *menu* screen and the *view* screen. The menu screen has command options and a disk directory. The view screen shows your 3D drawing. Shortly after the menu appears, the colors will alter and the program will switch to the view screen and display a simple 3-D object called "Plane" which is similar to *Figure 1.*

THE COMMANDS

With the exception of [D], any key pressed will take you to the view screen. One-key commands are acted upon immediately. No [RETURN] is needed.

[B] Returns you to the menu at any time.

[G] Switches you between GR. 7+ and GR. 8. GR. 7+ offers four colors (counting the background), changed with the [C] selection (below).

[C] Alters the GR. 7+ color registers. The message line at the bottom will indicate the current color number (0–3), its current hue and luminence values, plus the word Default.

The keys **[C]**, **[H]**, **[L]** increment the color, hue, and luminence. This can be used while in GR. 8. But the effect may be misleading because the GR. 7+ registers and the current screen registers are being altered but not the GR. 8 default values. The **[D]** key resets all GR. 7+ registers to default values. These values are updated each time you load a data file. Also, none of your playing will affect the menu screen colors, since these are main-

continued on next page

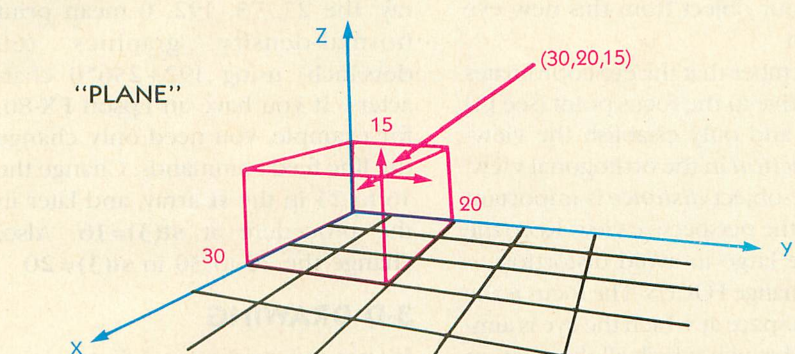


Figure 1

tained separately. Any other key terminates this routine.

[M] Magnifies the object. This is initially set at 4 and wraps to 1 when incremented past 9. You won't see the effect until the picture is redrawn by pressing [SPACE].

[V] Changes the view between perspective and orthogonal. Perspective, which emulates our vision, takes into account the distance from the eye, whereas orthogonal is used in drafting and engineering.

Figures 2 and 3 show the difference between perspective and orthogonal projections.

PREROTATE

These selections let you rotate your object about any of the three X, Y and Z axes. The message line shows the values of **rx** (rotate X), **ry** (rotate Y), **rz** (rotate Z), and **ri** (rotational increment). Each time the [X], [Y] or [Z] keys are pressed, the object rotates about the chosen axis in **ri** increments. The rotations are about axes that pass through the focus point.

The [I]/[J] keys increment/decrement the value of **ri** in degrees. Negative values of **ri** make rotations go in the opposite direction.

POINT OF VIEW

The following commands affect your dimensional view of the object.

[3] Fix EYE/dist. The eye coordinates are controlled by your joystick. Selections [1]-[4] use the same joystick scheme: Left/right alters the X coordinate, up/down alters the Y, and up/down while holding the trigger alters the Z. In selection [3], left/right with the trigger pressed controls the distance. Press [SPACE] to draw your object from this new eye location.

Remember that the eye coordinates are relative to the focus point (see [4] below) and only establish the viewing *direction* in the orthogonal view. The eye-object *distance* is important only in the perspective view. Keep the distance large to avoid distortion.

[4] Change FOCUS. The focus is the point in space at which the eye is aimed and through which all the rotation axes pass. It is normally on or near the object being studied and will be mapped to center screen (**cx,cy**). Move the flashing dot with your joystick. More importantly, watch its coordinates. Use [SPACE] to set your choice.

[5] Change CENTER. This alters **cx** and **cy**, shifting the object. These are actual screen coordinates (0,0 is the upper left). Use [SPACE] to set your choice and see the effect.

[0] Resets the center, eye, focus, magnification, and prerotation values to defaults used at start-up.

I/O

[D] Lists up to 22 data files in the menu window, assuming they have ".V3D" extenders. This is also done automatically at start-up and after each successful save.

[L] Loads a data file from disk. Answer the input prompt with a filename only. The program supplies the "D:" prefix and a ".V3D" extender. Upon hitting [RETURN] you'll see the full filespec. Press [L] again to accomplish the load. Any other key will abort the process.

[S] Saves data to a disk file. The process is the same as the above [L] load.

[P] Outputs to your printer. After pressing [P] you may choose to print the picture data [D] or the picture [P]. The picture is produced by a short screen dump for a Gemini 10X. You'll get best results by printing the GR. 8 picture.

To alter the printout procedure for your own printer, examine the **Prnt** procedure in the PRINTIO.ACT file and adapt accordingly. The **st** array contains printer control codes 26, 51, 16 which, on the Gemini, set the line feed to 16/144 inches. In the **pre** array, the 27, 75, 192, 0 mean print normal-density graphics (60 dots/inch) using 192+256*0 characters. If you have an Epson FX-80, for example, you need only change the line feed commands: Change the 16 to 24 in the **st** array, and later in the procedure at **st(3)=16**. Also, change the 20 to 30 in **st(3)=20**

3-D DRAWING

It's not easy to draw in 3 dimensions. The easiest way to learn is simply to try it. Concentrating on the changing coordinates in the message line may be easier than watching the dots and lines on the screen.

However, before you start, you may wish to save the object currently in memory. The process is easier to understand if you use the EDIT command, [2], to display a blank screen. Each time you press the [SPACE] bar, the screen will step through the drawing process of the object in memory, showing you how to construct a drawing.

To get started on your own, press [0] to use default values. To create a blank screen, press [2] then [1]. The joystick moves a flashing dot, whose coordinates appear in the bottom line. Position the cursor where you want it, and establish that point by pressing the [SPACE] bar. Your current updated point number will be displayed in the bottom line. Next, move the cursor to your second location, press [P] to switch from "Plot" to "LineTo" and press [SPACE] to draw the line.

For starters, keep it simple, or try editing a sample drawing. (The program can take up to 200 data points.) To edit a previous drawing, press [1] to ADD points and lines. As you step through the drawing by pressing [SPACE], you can change any of the values at any point, or you can begin adding to the last points. You can, of course, save your object to disk at any time.

DATA STORAGE

At this point, you need to understand a little about how data for your 3-D object is stored. The INTeger array **P** contains all the information in the following format:

P = [n:x y z:d:x y z:c: . . . . . :x y z:c: . . . . .]

P(0)=n is the number of data points in your object. The next four integers contain EYE data. The first three indicate the direction away from the FOCUS, and the fourth gives the distance.

The following four integers contain the three space coordinates of the focus point and a presently unused

value. These nine integers are followed by n data sets for your object. Each is made up of four integers containing the three space coordinates for a point and a fourth coded message. The encoding of the fourth integer is given by c= color + 16*p, where p=0 for "LineTo" and p=1 for "Plot".

SAMPLE DATA

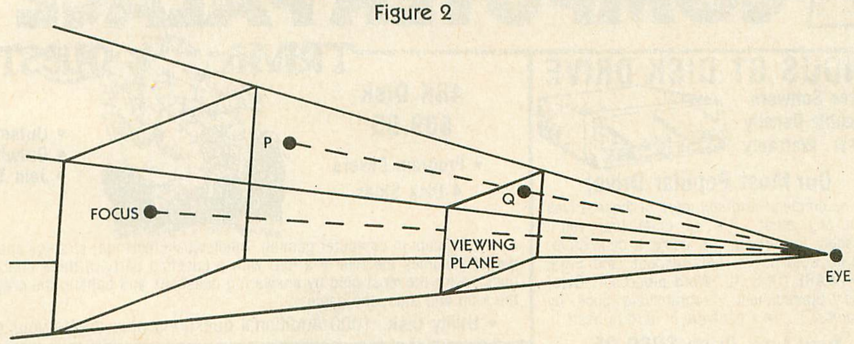
You can enter the data in figure 4 in the ADD mode to create a house with windows and a red chimney. Press [2], then [1] to clear memory. Now use your joystick to get the coordinates in the message line to match those of the first point in the example. Hit [C] and [P] as needed and set the data by hitting [SPACE]. Now do the same for the second point in the example, etc.

You can even do a little at a time. Just save the portion you've done. Next time load in this file, press [1] and continue from where you left off. A couple of examples will show you how to read the notation. "10 20 15:P2" means to Plot (10,20,15) in color 2. Whereas "20 20 30:L3" denotes a color 3 LineTo (20,20,30). Each example has suggested EYE and FOCUS data.

Longtime **Antic** *contributor Paul Chabot is a professor of mathematics and computer science at California State University, Los Angeles. He wrote "Splash In ACTION!" in our April 1985 issue.*

Listing on page 54.

PERSPECTIVE PROJECTION
Figure 2



ORTHOGONAL PROJECTION
Figure 3

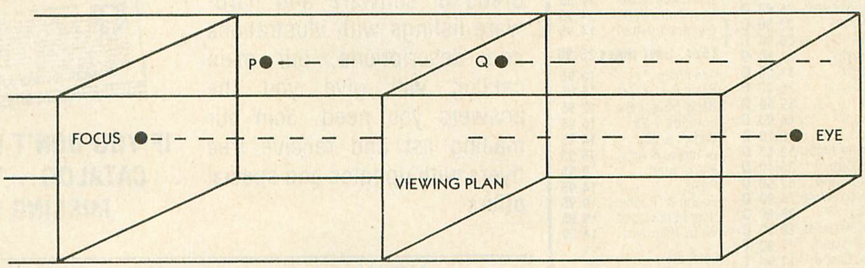


Figure 4

```
HOUSE : eye=(10 3 5:160) focus=(15 30 20)
(0 0 0:P2)      (30 0 0:L2)     (30 60 0:L2)
(0 60 0:L2)     (0 0 0:L2)      (0 0 40:L2)
(15 0 50:L2)    (30 0 40:L2)    (30 0 0:L2)
(30 60 0:P2)    (30 60 40:L2)   (15 60 50:L2)
(0 60 40:L2)    (0 60 0:L2)     (0 60 40:P3)
(0 0 40:L3)     (30 0 40:P3)    (30 60 40:L3)
(15 60 50:P2)   (15 0 50:L2)    (30 10 0:P3)
(30 10 25:L3)   (30 20 25:L3)   (30 20 0:L3)
(30 30 10:P3)   (30 30 25:L3)   (30 50 25:L3)
(30 50 10:L3)   (30 30 10:L3)   (30 40 10:P3)
(30 40 25:L3)   (10 60 10:P3)   (10 60 25:L3)
(20 60 25:L3)   (20 60 10:L3)   (10 60 10:L3)
(10 0 0:P1)     (10 -5 0:L1)    (20 -5 0:L1)
(20 0 0:L1)     (20 0 55:L1)    (20 -5 55:L1)
(20 -5 0:L1)    (10 0 0:P1)     (10 0 55:L1)
(10 -5 55:L1)   (10 -5 0:L1)    (10 0 55:P1)
(20 0 55:L1)    (10 -5 55:P1)   (10 -5 55:L1)
```

TURBO TYPO II

Speedy three-line enhancement

by DAVID McLAUGHLIN

*Changing three lines in TYPO II gives assembly language speed to **Antic's** program typing checker. (See Listing Section.) TURBO TYPO II will work on all Atari computers of any memory configuration.*

*When Andy Barton started working on TYPO II, he wrote it as a machine language vertical blank interrupt routine which fit in Page Six of memory. Following a suggestion from Bill Wilkinson of Optimized Systems Software—who wrote the original TYPO—we decided to switch to an all-BASIC TYPO II. **Antic** felt that novices, who most needed TYPO II, would have a much easier time if they didn't have to type in a lot of data statements.*

*We also wanted a simple program that the widest range of readers could have fun tinkering with. (See the I/O pages in both the April and May issues for some earlier enhancements.) Judging by the many thank-you letters from new Atari users and the large number of improvements that more experienced readers have sent in, **Antic** certainly succeeded in both goals.*—ANTIC ED

TURBO TYPO II

TYPO II is an enormous help, but it does have an annoying drawback—the longer the line, the more time required to check it. My object was to speed up TYPO II while producing the same two-letter code checksums.

I noticed the slowdown came from line 32150 of the original code. This line was an elegant solution, creating a unique code that also checks for correct letter order (i.e. distinguishes between TO and OT).

Therefore, the best way to speed up TYPO II was to write the line 32150 procedure in machine language. Thus, I created Listing 3, TYPO II, in ML. But you do NOT need to type this listing to create an enhanced TYPO II. It is included primarily for your information.

TWO CHOICES

I have provided two ways to create the necessary enhancements. Listing 1 is all you need to add to the current TYPO II. If you are one of those brave souls who doesn't mind typing in a lot of special and inverse characters then simply use your old TYPO II to type in Listing 1. When done, press [BREAK] then SAVE or LIST your enhanced TYPO II to disk or cassette.

If you'd rather not take a chance on typing tricky characters, then use your old TYPO II and follow these instructions in exactly the following order.

1. With TYPO II running, type in lines 32015 to 32025 from Listing 1.
2. Type in Listing 2.
3. After Listing 2 is successfully entered, press [BREAK] to stop TYPO II.
4. Type RUN and line 32026 from Listing 1 will be created for you.
5. When the READY prompt appears type GOTO 32000.
6. Type in line 32150 from Listing 1 and your enhanced TYPO II is complete.
7. Press [BREAK] and save the new TYPO II to disk or tape by typing:

LIST "D:TYPO II",32000,32220 or
LIST "C:",32000,32220.

If you want a SAVEd version then type NEW and then ENTER the new, enhanced TYPO II and then SAVE it to disk or tape.

SOURCE CODE

Listing 3 was created with the **Atari Assembler/Editor** cartridge. Of greatest interest is the use of three bytes to calculate TYPO II's variable ANS.

Normally, in this type of application, the programmer returns the value of ANS back to BASIC through memory locations 212 and 213 (hexadecimal \$D4 and \$D5) as the Atari creators originally planned. However, ANS can only have a maximum value of 65535, the greatest value two bytes can hold.

continued on next page

The original TYPO II's design allows ANS to become larger than that after the length of LINE$ becomes greater than about 20–30 characters. But the use of three-byte arithmetic is sufficient to hold the highest possible values of ANS.

A note of caution: TYPO II in ML uses three Page Six memory locations, 1789–1791 (hexadecimal $6FD-$6FF). Any program that accesses these three locations will be in conflict with the enhanced TYPO II. But since they are the last three locations on Page Six, there should be little problem.

Listing on page 75

G.U.P. the GREAT

Fast graphics power from BASIC

by DAREK MIHOCKA

Speed up your BASIC graphics commands and add powerful new ones—including circles, squares and patterned fills. Atari 400/800 owners can now access graphics commands available only on the XLs. Requires a disk drive and will run on all Atari computers with 48K memory.

Graphics Utility Package (G.U.P.) is a BASIC program that creates a boot file. When this file is installed in your Atari's memory several new and powerful graphics commands are available to you in the form of USR calls.

G.U.P. features include:

- PLOT and DRAWTO about 3–5 times faster than normal.
- Circles and boxes drawn at high speed.
- Several lines drawn at once for high speed.
- Access by a 400 or 800 to the 4 new XL graphics modes.
- Choose solid or multicolor patterns.
- Mix text with graphics.
- A simple 128 color rainbow command.
- Direct commands in GTIA modes.

TYPING G.U.P.

Type in Listings 1 and 2. Check them carefully with TYPO II and make sure no lines have been missed. These programs use machine language routines which could lock up your computer if mistyped, so be sure to SAVE backup copies before RUNning them.

Listing 1 will create an AUTORUN.SYS file on disk. RUN the program and at the prompt insert a formatted disk with DOS 2.0S, press START and G.U.P. will be written to disk. (Note: because of the G.U.P. file structure, do NOT use DOS 3.). Now,

continued on next page

boot the disk, the screen will change color and G.U.P. will be in memory.

DEMONSTRATION

To test G.U.P., LOAD Listing 2 into memory and RUN it. Listing 2 is a demonstration program that will take G.U.P. through its amazing paces and provide examples of how to use G.U.P. in your own BASIC programs.

Lines 40–180 of Listing 2 are essential to any BASIC program using the G.U.P. commands. These lines determine the starting locations of the different routines and store them in command variables. You can renumber them for your convenience, but they must be executed before any G.U.P. commands are given.

LOCK-UP

G.U.P. is really a series of USR calls, which are assembly language routines that are accessed from BASIC. Computer newcomers, who may be a little wary of USR routines should read last month's "USR Routines" by Ernie Negus.

When using assembly language routines, there is always a good possibility of computer lock-up—your keyboard no longer responds and there's nothing to do but turn off your computer and start all over again. Lock-ups can occur with just *one* mistyped character. They're time-consuming and annoying and you may lose unsaved data. But they won't harm your machine, so don't be afraid to experiment.

G.U.P. COMMANDS

G.U.P. has 10 separate commands: GRAPHICS, SET, PLOT, DRAWTO, LINE, BOX, CIRCLE, TEXT, RANDOM, and C128. Here is a description of each:

GRAPHICS x—Establish graphics mode. Any one of the 16 graphics modes can be selected. This allows a ROM B Atari to access the 4 new ROM C graphics modes with one command. To access the 160X192 4 color mode (GR.15 on XL computers), use the command:

A = USR(GRAPHICS,15)

SET a,b,c,d—Choose patterns. Each parameter contains color data for 4 pixels necessary for the pattern of colors. (See the explanation later on.) To set the colors to simulate the BASIC command COLOR 1, use:

A = USR(SET,85,85,85,85)

PLOT x,y—Same as BASIC's PLOT. Use:

A = USR(PLOT,X,Y)

DRAWTO x,y—Again, the same as BASIC's DRAWTO. Use:

A = USR(DRAWTO,X,Y)

LINE x1,y1,x2,y2—Similar to DRAWTO, except that the first parameter is the starting pixel. For example, to draw lines from pixel 0,0 to 10,5 to 20,3 use:

A = USR(LINE,0,0,10,5,20,3)

The DRAWTO and LINE commands can have more than one parameter. That is, if 5 pairs of coordinates are given, then a line will be drawn from the first to the second, then the second to third, and so on. This saves time and memory.

BOX x1,y1,x2,y2—This will draw a filled-in box whose opposite corner coordinates are (x1,y1) and (x2,y2). A 10 pixel square box is draw with:

A = USR(BOX,0,0,10,10)

CIRCLE x,y,r—this draws a circle of radius r pixels at location x1, y1. This command works about 30 times faster than any BASIC algorithm I've tried. A large circle in GR.15 is drawn by:

A = USR(CIRCLE,79,79,75)

TEXT x,y,"abcd..",l—this will print out the given string of length l in graphics modes 4,6 or 8 at coordinates x,y. This allows easy mixing of text and graphics. It should be used in a 2 color mode to make the letters readable. To put the word ANTIC in the upper left corner of the screen use:

A = USR(TEXT,0,0,ADR ("ANTIC"),5)

Repeating the command will erase the word and restore any graphics that were underneath.

RANDOM—This command simply puts random colors in the color pattern. The command is:

A = USR(RANDOM)

C128 d,r—This produces a 128 color rainbow display for the given duration (d in 1/60ths of a second) in the given color register (r). R is determined by subtracting 704 from the location of the color register. For example, to make a rainbow of the background (location 710) for 10 seconds use:

A = USR(C128,600,6)

VARIABLE REGISTERS

Four variables are established as registers in G.U.P.

CIRF is a variable used to switch between empty-circle drawing and filled circles. POKE CIRF,0 for empty

circles or POKE CIRF,1 for filled ones.

Memory location 208 is used with the command GRAPHICS. Since G.U.P. can only plot up to a co-ordinate of 255,191, the co-ordinates from 256,191 to 319,191 in graphics mode 8 cannot normally be accessed. By poking an 8 into 208, the co-ordinate plane is shifted over 64 pixels and A=USR(PLOT,0,0) would actually plot at 64,0.

The memory locations SETCOLOR to SETCOLOR+8 are values that are copied into memory locations 704 to 712 when a GRAPHICS command is executed. By poking into these locations, the default colors of the screen can be changed, as they already are in G.U.P. For example, in graphics mode 0, to change the screen color to black, a POKE 710,0 is run. To set G.U.P. to make the screen black every time a GRAPHICS command is given, do a POKE SETCOLOR+6,0 since 710=704+6.

SET COMMAND

One of the best things about G.U.P. is that it doesn't just draw in solid colors. A 1x4, 2x4, 4x4 or 8x4 color pattern is stored at memory locations COLOR to COLOR+3. The size of the pattern is determined by the graphics mode. This is because text modes use one byte per character, GTIA modes use one byte to display 2 pixels, 4-color modes use one byte for 4 pixels, and the 2-color modes can store 8 pixels in a byte.

The pattern can either be POKEd in, or put in with the SET command. This way circles can be drawn in a checkerboard pattern, lines can be dotted, and more.

The formula for determining the parameters to use in the SET command is quite simple. Just like BASIC's COLOR command which selects one of the 4 color registers for plotting, each one of the 4 parameters passed in the SET command determines the color registers for a 4×1 array of pixels in a 4-color mode, 8×1 in a 2 color mode, 2×1 in a GTIA mode.

Using 2-color modes 4, 6 and 8, an 8x4 pattern can be set up. This means that during PLOT, DRAWTO, LINE, BOX, or CIRCLE, the pixels will be plotted so that if the whole screen is filled up it will be made up of little 8x4 boxes of the same pattern. By making the pixels alternate from black to white, a very fine checkerboard pattern is made as follows: Let 0 represent a pixel in color 0 and a 1 represent a pixel in color 1. The pattern is therefore

```
01010101
10101010
01010101
10101010
```

This pattern represents 4 binary numbers. In decimal they are 85, 170, 85 and 170. Therefore the command to give is:

```
A=USR(SET,85,170,85,170)
```

If you just want to plot with color 1, the whole pattern would be 1's and the four numbers would each be 255 (11111111 in binary). Similarly, if you want to plot with color 0, use the number 0 four times.

In a 4-color mode, 4 colors can be used so each pixel must be represented by two bits: 00, 01, 10 or 11. This is why only 4 can be used in one byte. Similarly, in a GTIA mode, 16 colors are possible, therefore 4 bits are required per pixel and only 2 pixels fit in one byte.

In a text mode, 0, 1, 2, 12 or 13, the four numbers represent a block of 1×4 characters. Each number is the ATASCII code of the character.

If you wish to POKE these values instead of using the SET command, locations COLOR to COLOR+3 are where the four parameters of the SET command go and can be POKEed directly.

FEEL FREE

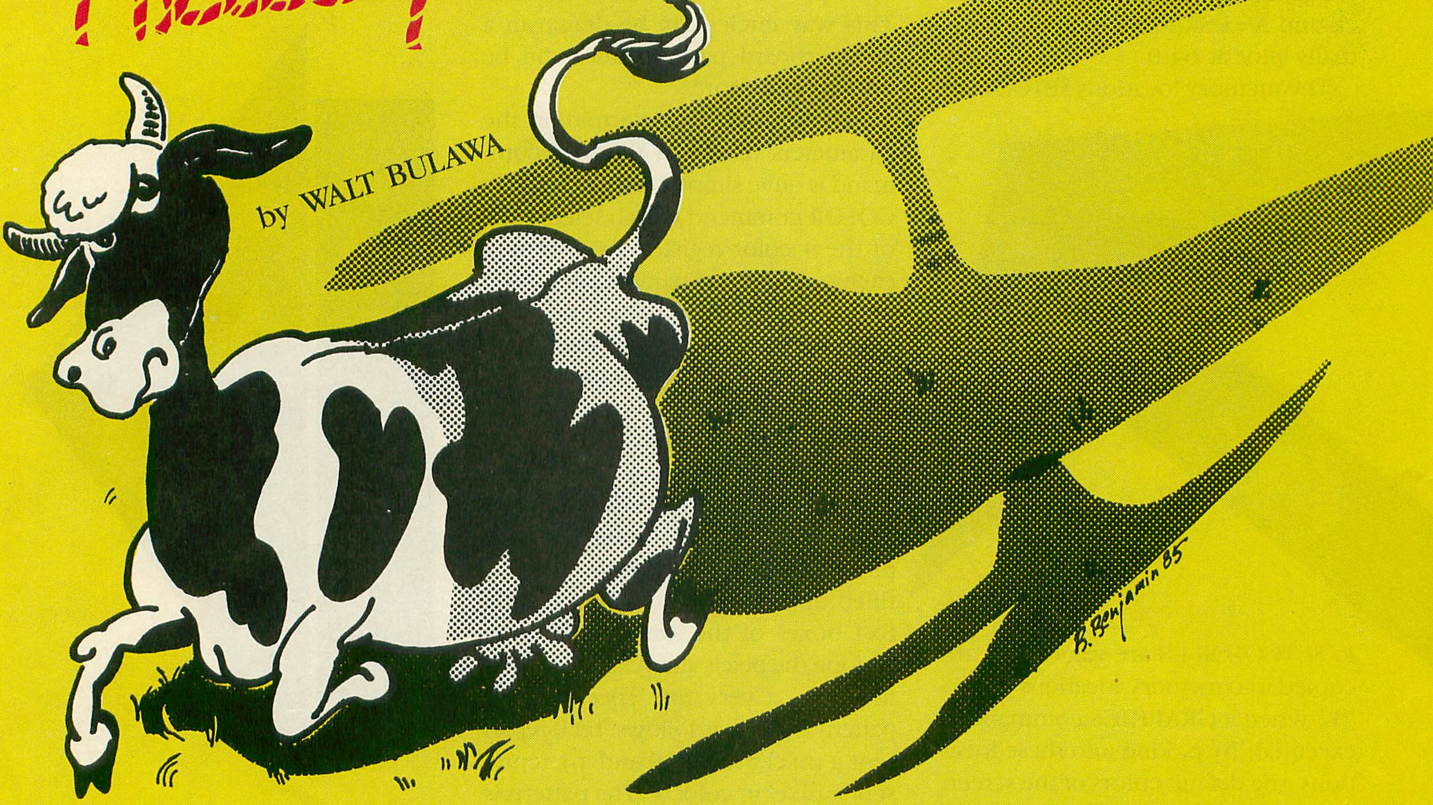
Luckily, you don't really need to understand any of this. Simply try out different numbers in the SET parameters until you see what you like. This holds true of all of G.U.P. Many of these routines may be placed in strings for those more advanced programmers. Feel free to experiment.

Darek Mihocka placed in the top 10 in three nationwide Canadian university math and physics contests. He's a member of the Toronto Atari Federation and a licensed glider pilot.

Listing on page 68

Helicopter Round-up

by WALT BULAWA



In Australia there are more men than women and more cows than men. To keep the cows under control, ranchers use helicopters. Your helicopter is supposed to scare the cows into their holding pen.

In case you've never buzzed a bovine with a helicopter before, you'll need some advice. As you maneuver your joystick-controlled copter closer to the herd; the animals get increasingly skittish and move away from you. Steering heifers into a corral isn't easy.

While patiently waiting for you to get airborne, the cows move randomly. Then after all of the cattle have been packed into their pen, the gate will close. You must return the copter to the landing pad for a full score.

But if you run out of fuel, your helicopter will crash and kill some innocent cows.

USING THE PROGRAM

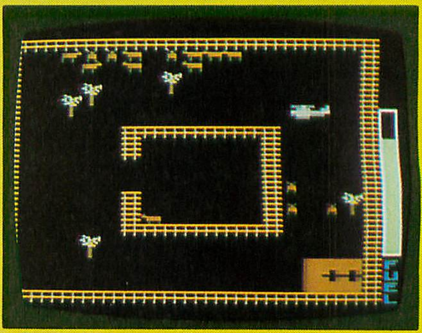
To get this game off the ground, type in Listing 1, check it with TYPO II, and SAVE a copy. When you RUN Helicopter Roundup, you'll have to wait while the screen blanks for initialization (don't panic). Then you'll see the game.

You can [SELECT] the number of cows to herd, and you can refuel during play by positioning the copter over the landing pad and pushing the joystick trigger.

The game continues indefinitely until all cows are controlled or you crash.

When he's not rounding up redefined cow characters, Walt Bulawa is an application programmer for the medical field.

Listing on page 71



Cows? Helicopters? In your Atari? You bet! This BASIC action game challenges you to round 'em up from the air. Works on all Atari computers of any memory configuration. Disk or cassette.

continued from page 37

A dotted half note gets held for three beats of a measure. Sustained notes slide into the note which follows them instead of playing as separate notes.

To choose a dotted, sustained eighth note, you would press the following three keys—[.] [8] [S]. The Musician isn't fussy. You may press keys in any order for a multiple-key note command.

When you use the [#] or [B] to make sharps or flats, The Musician will stay in the sharp or flat mode until you press [N] for Natural.

Rests are times when no note is played. To choose a quarter rest, first press [Q] to make a quarter note, then press [R] to make it a rest. Similarly, to make an eighth rest, press [8] [R]. Half rests are [H] [R].

Once you have selected the correct type of note and placed it in the right position on the staff, press your joystick button to enter it.

Press [C] to Cancel the note if you change your mind. [C] always cancels the last note or the last chord entered.

ENTERING CHORDS

The Musician will recognize even the most advanced kinds of chords. Entering chords is a snap since you enter them by name, not by notes. Press the asterisk [*] key to tell The Musician you want to enter a chord. The Musician will display:

ENTER A CHORD
THEN PRESS RETURN

To enter a C chord, press [*] followed by [C], then press [RETURN]. The Musician will display the name of the chord you have keyed in, and briefly play the chord.

Using the Key of C as an example, here is a complete list of chord types recognized by The Musician.

ENTER	CHORD SELECTED
C	C major
CM	C minor
CM7	C minor seventh
C7	C seventh
CM6	C minor sixth
C6	C sixth
CMAJ	C major seventh
C+	C augmented fifth
C-	C diminished fifth
CD	C diminished

Sharp and flat chords are entered by adding the [#] or [B] keys. For instance, here's how to enter a C sharp minor seventh chord:

Press [*]
Press [C]
Press [#]
Press [M]
Press [7]
Press [RETURN]

Chords in other keys are entered in exactly the same manner. For instance, you enter a G seventh as G7.

If a chord is to play simultaneously with a certain note, *enter the chord* first, then enter the note.

If you make a mistake when entering a chord and wish to exit from chord mode, simply press [*] again. The chord will not be entered and you will return to note entry mode again. To re-enter the chord, press [*] again.

Chords will continue to play in the background until they are changed to something else or turned off. To turn off a background chord, press [O]. The Musician will display: CHORD OFF.

SONG PLAYBACK

To hear a playback of the song you have entered, press [P]. The song will play automatically and each note will appear on the staff as it plays. You may change the tempo of the song by pressing [T] before pressing [P]. The Musician will display:

CHANGE TEMPO SLOWER

Press the joystick button to slow down the tempo. To speed up the tempo, first move the joystick forward. The word SLOWER will change to FASTER. Pressing the joystick button will then speed up the tempo.

After playing a song, it remains in the computer memory. You may continue adding notes to the end of the song. You may clear it from memory. Or you may SAVE it to cassette or disk.

Pressing [ESC] while a song is playing causes The Musician to immediately stop playing the song and return to note entry mode.

SAVING A SONG

To SAVE a song, press [A] for Archive. The Musician will display:

ENTER SONG NAME

If you are saving to cassette, just press [RETURN]. You will hear two beeps. Put your recorder on record, then press [RETURN].

If you are using disk, type in the song name. Do not use more than eight characters, since this becomes the file name. Three-character extensions are allowed if you like using them. Press [RETURN]. The Musician will write out your song.

LOADING A SONG

To load a previously saved song, press [L]. The Musician will ask you for the name of the song file.

If you are entering the song from cassette, position the tape to the beginning of the song and press [RETURN].]

If you are using disk, type in the song name and press [RETURN].

The Musician will load in the song and play it for you. Once a song is loaded it remains in memory until you clear it out. If you add notes after loading a song, they get appended to the end of the song. In this way, you can finish songs you saved only partially completed.

CLEAR AND EXIT

To clear a song from memory, press [CLEAR]. Do NOT use the [SHIFT] key. Now you may begin entering a new song.

To exit The Musician, press [E]. The Musician will display

EXIT Y or N

Press [Y] to confirm your choice. The program will end and return you to BASIC.

*Angelo Giambra is a senior programmer for Marine Midland Bank in Rochester, N.Y. His "Eight Queens Problem" appeared in the April, 1984 **Antic**.*

Listing on page 61

SOFTWARE LIBRARY

Antic type-in listing section includes every full-length program from this issue. Listings are easier to type and proofread, easy to remove and save in a binder if you wish.

DISK SUBSCRIBERS: You can use all these programs immediately. Just follow the instructions in the accompanying magazine articles.

Antic program listings are typeset by Star's Gemini 10X Printer—From Star Micronics, Inc., 200 Park Avenue, New York, NY 10166.

TYPING SPECIAL ATARI CHARACTERS

Antic printed program listings leave a small space between each Atari Special Character for easier reading. Immediately below you will see the way **Antic** prints all the standard Atari letters and numbers, in upper and lower case, in normal and inverse video.

ABCDEFGHIJKLMNOPQRSTUVWXYZ

abcdefghijklmnopqrstuvwxyz

0123456789

The Atari Special Characters and the keys you must type in order to get them are shown in the two boxes below. (Squares are drawn around the normal video characters so you can see their positions more accurately, these squares will not appear in listings.)

NORMAL VIDEO

FOR THIS	TYPE THIS	FOR THIS	TYPE THIS
	CTRL ,		CTRL T
	CTRL A		CTRL U
	CTRL B		CTRL V
	CTRL C		CTRL W
	CTRL D		CTRL X
	CTRL E		CTRL Y
	CTRL F		CTRL Z
	CTRL G		ESC ESC
	CTRL H		ESC CTRL -
	CTRL I		ESC CTRL =
	CTRL J		ESC CTRL +
	CTRL K		ESC CTRL *
	CTRL L		CTRL .
	CTRL M		CTRL ;
	CTRL N	|	SHIFT =
	CTRL O		ESC SHIFT CLEAR
	CTRL P		
	CTRL Q		
	CTRL R		ESC DELETE
	CTRL S		ESC TAB

INVERSE VIDEO

FOR THIS	TYPE THIS	FOR THIS	TYPE THIS
	Λ CTRL ,		Λ CTRL Y
	Λ CTRL A		Λ CTRL Z
	Λ CTRL B		ESC SHIFT DELETE
	Λ CTRL C		
	Λ CTRL D		
	Λ CTRL E		ESC SHIFT INSERT
	Λ CTRL F		
	Λ CTRL G		
	Λ CTRL H		ESC CTRL TAB
	Λ CTRL I		
	Λ CTRL J		
	Λ CTRL K		ESC SHIFT TAB
	Λ CTRL L		
	Λ CTRL M		
	Λ CTRL N		Λ CTRL .
	Λ CTRL O		Λ CTRL ;
	Λ CTRL P		Λ SHIFT =
	Λ CTRL Q		ESC CTRL 2
	Λ CTRL R		ESC CTRL DELETE
	Λ CTRL S		
	Λ CTRL T		
	Λ CTRL U		ESC CTRL INSERT
	Λ CTRL V		
	Λ CTRL W		
	Λ CTRL X		

Whenever the CONTROL key (CTRL on the 400/800) or SHIFT key is used, ***hold it down*** while you press the next key. Whenever the ESC key is pressed, ***release*** it before you type the next key.

Turn on inverse video by pressing the Reverse Video Mode Key. Turn it off by pressing it a second time. (On the 400/800, use the Atari Logo Key instead.) Note: In the printed listings, inverse characters willl be slightly smaller than the normal ones.

Among the most common program typing mistakes are switching certain capital letters with their lower-case counterparts—you need to look especially carefully at P, X, O and 0 (zero).

Some of Atari Special Characters are not easy to tell apart from standard alpha-numeric characters. Usually the Special Characters will be ***thicker*** than the alpha-numerics. Compare the two sets of characters below:

SPECIAL		STANDARD	
╱	CTRL F	/	/
╲	CTRL G	\	SHIFT +
—	CTRL N	—	SHIFT -
—	CTRL R	-	-
+	CTRL S	+	+

HOW TO USE TYPO II

Type in TYPO II and SAVE a copy to disk or cassette.

Type GOTO 32000 and follow TYPO II onscreen instructions. If the resulting two-letter line codes are not exactly the same as those in the magazine, you mistyped something in that line.

To call back any line previously typed, type an asterisk [*] followed (without in-between spaces) by the line number, then press [RETURN]. When the complete line appears at the top of the screen, press [RETURN] again. This is also the way you use TYPO II to proofread itself.

To LIST your program, press [BREAK] and type LIST. To return to TYPO II, type GOTO 32000.

To remove TYPO II from your program, type LIST "D:FILENAME",0,31999 [RETURN] (Cassette owners LIST "C:). Type NEW, then ENTER "D:FILENAME" [RETURN] (Cassette—ENTER "C:). Your program is now in memory without TYPO II and you can SAVE or LIST it to disk or cassette.

Owners of the BASIC XL cartridge from O.S.S. type SET 5,0 and SET 12,0 before using TYPO II.

```
WB 32000 REM TYPO II BY ANDY BARTON
VM 32010 REM VER. 1.0 FOR ANTIC MAGAZINE
HS 32020 CLR :DIM LINE$(120):CLOSE #2:CLO
   SE #3
BN 32030 OPEN #2,4,0,"E":OPEN #3,5,0,"E"
YC 32040 ? "↖":POSITION 11,1:? "TYPO II"
EM 32050 TRAP 32040:POSITION 2,3:? "Type
   in a program line"
HS 32060 POSITION 1,4:? " ":INPUT #2;LINE
   $:IF LINE$="" THEN POSITION 2,4:LIST B
   :GOTO 32060
XH 32070 IF LINE$(1,1)="*" THEN B=VAL(LIN
   E$(2,LEN(LINE$))):POSITION 2,4:LIST B:
   GOTO 32060
TH 32080 POSITION 2,10:? "CONT"
MF 32090 B=VAL(LINE$):POSITION 1,3:? " ";
NY 32100 POKE 842,13:STOP
CN 32110 POKE 842,12
ET 32120 ? "↖":POSITION 11,1:? " TYPO II
   ":POSITION 2,15:LIST B
CE 32130 C=0:ANS=C
QR 32140 POSITION 2,16:INPUT #3;LINE$:IF
   LINE$="" THEN ? "LINE ";B;" DELETED":G
   OTO 32050
VV 32150 FOR D=1 TO LEN(LINE$):C=C+1:ANS=
   ANS+(C*ASC(LINE$(D,D))):NEXT D
WJ 32160 CODE=INT(ANS/676)
JW 32170 CODE=ANS-(CODE*676)
EH 32180 HCODE=INT(CODE/26)
BH 32190 LCODE=CODE-(HCODE*26)+65
HB 32200 HCODE=HCODE+65
IE 32210 POSITION 0,16:? CHR$(HCODE);CHR$
   (LCODE)
VG 32220 POSITION 2,13:? "If CODE does no
   t match press  RETURN  and edit line a
   bove.":GOTO 32050
```

ERROR FILE

BUS OVERLINES

Some signals and address labels were printed without overlines in Part III of Earl Rice's Parallel Bus Revealed" (Antic, March 1985).

These are the correct labels:

$\overline{\text{D8XX-DFXX}}$
$\overline{\text{CS}}$ (CHIP RESET)
R/$\overline{\text{W}}$
$\overline{\text{D1XX}}$
$\overline{\text{RDE}}$ (READ DATA ENABLE)
$\overline{\text{DS}}$ (DATA STROBE)
$\overline{\text{DRST}}$ (DEVICE RESET)

FIRST LESSON IN ASSEMBLY

Line 100 of the listing for "First Lesson in Assembly Language" (November, 1984) should read POKE 755,4 instead of POKE 775,4.

KOOKY'S QUEST

February '85

The following line is missing:
2100 FOR S=32 TO 16 STEP -4: SOUND 0,S,14,10: EA=EA*EA*EA: SOUND 0,0,0,0: EA=1 ∧0:NEXT S

DRUM SYNTH

February '85

In Figure 1, the "ART" should be the Fuji (inverse) symbol.

MISSING INFOBITS

DECEMBER '84

The AL source listing for Infobits (Dec. '84) was left out of the previous issue. You'll find it in the Jan. '85 Software Library.

ADVENT X-5

November '84

Missing line: 8020 RUN. Also, cassette owners should change the 138 in line 4005 to 130. The TYPO II code for line 1005 is EJ.

ADVENTURE ISLAND

November '84

Line 837 is missing its last item of data, a 4. Also, it will not run with DOS XL.

rotate and zoom 3-D images in ACTION

VIEW 3-D

Article on page 38.

LISTING 1

```
; GR78M (LISTING 1)

MODULE:INT xnow=[80],ynow=[90]
BYTE cnow=[1],key=764,ram=106,cur=752
BYTE ARRAY mask7=[64 16 4 1],clor=708
  ,mask8=[128 64 32 16 8 4 2 1],mask,row
CARD dlist=560,sa=88
CARD ARRAY adr(192):CARD POINTER mes

PROC Kolor(BYTE c) RETURN
PROC Dot(INT x,y) RETURN
PROC Dit(INT x,y) RETURN

PROC LineTo(INT x,y)
INT dx,dy,xf,yf,a,b,t,i
Dot(xnow,ynow)
IF x=xnow AND y=ynow THEN RETURN FI
IF x>xnow THEN dx=x-xnow:xf=1
ELSE dx=xnow-x:xf=-1 FI
IF y>ynow THEN dy=y-ynow:yf=1
ELSE dy=ynow-y:yf=-1 FI
x=xnow:y=ynow
IF dx>dy THEN a=dy+dy:t=a-dx:b=t-dx
  FOR i=1 TO dx DO x==+xf
    IF t<0 THEN t==+a
    ELSE t==+b:y==+yf
    FI Dot(x,y)
  OD
ELSE a=dx+dx:t=a-dy:b=t-dy
  FOR i=1 TO dy DO y==+yf
    IF t<0 THEN t==+a
    ELSE t==+b:x==+xf
    FI Dot(x,y)
  OD
FI xnow=x:ynow=y:RETURN

PROC Gr78ON():BYTE i:BYTE ARRAY dl
Graphics(8+16):adr(0)=sa:dl=dlist
FOR i= 1 TO 191 DO adr(i)=adr(i-1)+40 OD
dl==-4:dl(0)=112:dl(1)=80:dl(2)=16
FOR i=3 TO 198 DO dl(i)=dl(i+4) OD
dl(199)=16:dl(200)=66:mes=dl+201
dl(204)==-4:dlist=dl:RETURN

PROC Kolor7(BYTE c):BYTE i
c==& 3:cnow=c
FOR i=0 TO 3 DO mask(3-i)=c:c==LSH 2 OD
RETURN

PROC Dot7(INT x,y):BYTE xb,xr
BYTE ARRAY pre=[63 207 243 252]
IF x<0 OR x>159 THEN RETURN FI
IF y<0 OR y>191 THEN RETURN FI
xb=x RSH 2:xr=x AND 3:row=adr(y)
row(xb)==& pre(xr) % mask(xr):RETURN

PROC Dit7(INT x,y):BYTE xb,xr
IF x<0 OR x>159 THEN RETURN FI
IF y<0 OR y>191 THEN RETURN FI
xb=x RSH 2:xr=x AND 3:row=adr(y)
row(xb)==! mask(xr):RETURN

PROC Gr7(BYTE ARRAY d):BYTE i
mask=mask7:Kolor=Kolor7:Dot=Dot7
Dit=Dit7:d(3)=78:d(99)=78
FOR i=6 TO 98 DO d(i)=14 OD
FOR i=102 TO 198 DO d(i)=14 OD:RETURN

PROC Kolor8(BYTE c):BYTE i
cnow=c & 3:IF c>1 THEN c=1 FI
FOR i=0 TO 7 DO mask(7-i)=c:c==LSH 1 OD
RETURN

PROC Dot8(INT x,y):BYTE xb,xr
BYTE ARRAY
  pre=[127 191 223 239 247 251 253 254]
IF x<0 OR x>319 THEN RETURN FI
IF y<0 OR y>191 THEN RETURN FI
xb=x RSH 3:xr=x AND 7:row=adr(y)
row(xb)==& pre(xr) % mask(xr):RETURN

PROC Dit8(INT x,y):BYTE xb,xr
IF x<0 OR x>319 THEN RETURN FI
IF y<0 OR y>191 THEN RETURN FI
xb=x RSH 3:xr=x AND 7:row=adr(y)
row(xb)==! mask(xr):RETURN

PROC Gr8(BYTE ARRAY d):BYTE i
mask=mask8:Kolor=Kolor8:Dot=Dot8
Dit=Dit8:d(3)=79:d(99)=79
FOR i=6 TO 98 DO d(i)=15 OD
FOR i=102 TO 198 DO d(i)=15 OD:RETURN
```

LISTING 2

```
; MISC1 (LISTING 2)

MODULE:BYTE st:INT ARRAY
  jx=[1 1 1 1 1 2 2 2 1 0 0 0 1 1 1 1]
  ,jy=[1 1 1 1 1 2 0 1 1 2 0 1 1 2 0 1]
BYTE ARRAY b="                        "

PROC Setjxjy(BYTE i)
FOR i=0 TO 15 DO jx(i)==-1:jy(i)==-1 OD
RETURN

PROC Pb(BYTE i):b(0)=i:Print(b):RETURN

PROC Wait(CARD w,j)
FOR j=0 TO w DO w==+1:w==-1 OD RETURN

; TRIG
MODULE:BYTE ARRAY Si(91)

PROC SetTrig(BYTE t INT y)
FOR t=0 TO 90 DO y=(t*t)/45
  y=(y*t)/5:y=100*t-y:y==/45:Si(t)=y
OD RETURN


INT FUNC sin(INT t,y):t==MOD 360
IF t<91 THEN y=Si(t)
ELSEIF t<181 THEN y=Si(180-t)
ELSEIF t<271 THEN y=-Si(t-180)
ELSE y=-Si(360-t) FI RETURN(y)

INT FUNC cos(INT t,y):t==MOD 360
IF t<91 THEN y=Si(90-t)
ELSEIF t<181 THEN y=-Si(t-90)
```

```
ELSEIF t<271 THEN y=-Si(270-t)
ELSE y=Si(t-270) FI RETURN(y)

; VECTOR
INT FUNC ABS(INT x)
IF x<0 THEN x=-x FI RETURN(x)

INT FUNC SQR(INT x):INT y
IF x=0 THEN RETURN(0) FI:x=ABS(x):y=0
DO y==+1:IF y*y+y>x THEN RETURN(y) FI OD

INT FUNC Vdot(INT ARRAY v,w):INT x
x=v(0)*w(0):x==+v(1)*w(1)
x==+v(2)*w(2):RETURN(x)

PROC Vprod(INT ARRAY v,w,u)
u(0)=v(1)*w(2):u(0)==-v(2)*w(1)
u(1)=v(2)*w(0):u(1)==-v(0)*w(2)
u(2)=v(0)*w(1):u(2)==-v(1)*w(0)
RETURN

PROC Normize(INT ARRAY v):INT i,j,s
i=ABS(v(0))
j=ABS(v(1)):IF i<j THEN i=j FI
j=ABS(v(2)):IF i<j THEN i=j FI
IF i>100 THEN j=1+i/100
  FOR i=0 TO 2 DO v(i)==/j OD
FI
FOR j=0 TO 1 DO s=Vdot(v,v):s=SQR(s)
  FOR i=0 TO 2 DO v(i)=v(i)*128/s OD
OD RETURN
```

LISTING 3

```
; COLORS (LISTING 3)

MODULE:BYTE ARRAY dfault(5),CP
  ,C7=[52 24 130 194 0]
  ,C8=[52 26 0 194 208]
  ,CM=[52 24 194 130 80]

PROC IncC():BYTE i
i=cnow:cnow==+1:Position(10,23)
IF i=3 THEN i=4:cnow=0 FI
Kolor(cnow):i=clor(i):PrintB(cnow)
Position(18,23):PrintB(i RSH 4)
Put(' ):Position(27,23)
PrintB(i & 14):Put(' ):RETURN

PROC IncHue():BYTE i,j
IF cnow=0 THEN i=4 ELSE i=cnow-1 FI
j=clor(i) RSH 4
j==+1:IF j>15 THEN j=0 FI
Position(18,23):PrintB(j):Put(' )
clor(i)=(j LSH 4)+(clor(i) & 14)
C7(i)=clor(i):RETURN

PROC IncLum():BYTE i,j
IF cnow=0 THEN i=4 ELSE i=cnow-1 FI
j=clor(i) & 14
j==+2:IF j>15 THEN j=0 FI
Position(27,23):PrintB(j):Put(' )
clor(i)=(clor(i) & 240)+j
C7(i)=clor(i):RETURN

PROC DfaultC():BYTE i
FOR i=0 TO 4 DO C7(i)=dfault(i)
  clor(i)=C7(i)
OD RETURN

PROC SetDfault():BYTE i
FOR i=0 TO 4 DO dfault(i)=C7(i) OD:RETURN

PROC CPon():BYTE i
FOR i=0 TO 4 DO clor(i)=CP(i) OD:RETURN

PROC CMon():BYTE i
FOR i=0 TO 4 DO clor(i)=CM(i) OD:RETURN

PROC FixCol():IncC()
DO WHILE key=255 DO OD
  IF key=18 THEN key=255:IncC()
  ELSEIF key=57 THEN key=255:IncHue()
  ELSEIF key=0 THEN key=255:IncLum()
  ELSEIF key=58 THEN key=255:DfaultC()
  ELSE EXIT FI
OD RETURN
```

LISTING 4

```
; DRAW3D (LISTING 4)

MODULE:BYTE vflag=[0],gflag=[7]
INT sx,sy,mag=[3],cx=[80],cy=[90]
 ,rx=[0],ry=[0],rz=[0],ri=[30]
CARD sa1,dl1,dl2,lin16,lin17,lin18
 ,lin19,lin20,lin21,lin22,lin23,lin15
INT ARRAY P(809),eye,foc,R(9),E(9),M(9)
 ,Q=[21:10 5 5:100:0 0 0:0:
  50 0 0:18:0 0 0:2:0 50 0:2:
  0 0 0:18:0 0 40:2:10 0 0:17:
  10 50 0:1:20 50 0:17:20 0 0:1:
  30 0 0:17:30 50 0:1:40 50 0:17:
  40 0 0:1:0 10 0:19:50 10 0:3:
  50 20 0:19:0 20 0:3:0 30 0:19:
  50 30 0:3:50 40 0:19:0 40 0:3]

PROC FixP(INT ARRAY Q):INT i,j
Zero(P,1618):j=4*Q(0)+8
FOR i=0 TO j DO P(i)=Q(i) OD
eye=P+2:foc=P+10:RETURN

PROC Rot(INT ARRAY v):INT x,y,z,s,c
y=v(1)
v(1)=y*cos(rx)/128:v(2)=y*sin(rx)/128
x=v(0):z=v(2):s=sin(ry):c=cos(ry)
v(0)=(x*c-z*s)/128:v(2)=(x*s+z*c)/128
x=v(0):y=v(1):s=sin(rz):c=cos(rz)
v(0)=x*c-y*s:v(1)=x*s+y*c:v(0)==/128
v(1)==/128:Normize(v):RETURN

PROC FixR():INT ARRAY v(3),w(3),u(3)
v(0)=128:v(1)=0:v(2)=0:Rot(v)
w(0)=0:w(1)=128:w(2)=0:Rot(w)
Vprod(v,w,u):Normize(u)
R(0)=v(0):R(1)=v(1):R(2)=v(2)
R(3)=w(0):R(4)=w(1):R(5)=w(2)
R(6)=u(0):R(7)=u(1):R(8)=u(2):RETURN

PROC FixE():INT s
E(6)=eye(0):E(7)=eye(1):E(8)=eye(2)
Normize(E+12)
IF E(8)=0 THEN E(3)=0:E(4)=0:E(5)=128
ELSEIF E(6)=0 AND E(7)=0 THEN
  E(3)=0:E(4)=128:E(5)=0
ELSE E(3)=-E(6):E(4)=-E(7)
  E(5)=E(6)*E(6):E(5)==+E(7)*E(7)
  E(5)==/E(8):Normize(E+6)
  IF E(8)<0 THEN E(3)=-E(3):E(4)=-E(4)
    E(5)=-E(5)
  FI
FI Vprod(E+6,E+12,E):Normize(E):RETURN

PROC FixM()
M(0)=Vdot(R,E):M(3)=Vdot(R,E+6)
M(1)=Vdot(R+6,E):M(4)=Vdot(R+6,E+6)
```

continued on next page

```
M(2)=Vdot(R+12,E):M(5)=Vdot(R+12,E+6)
Normize(M):Normize(M+6)
Vprod(M,M+6,M+12):Normize(M+12):RETURN

PROC Maksxsy(INT ARRAY v):BYTE i
INT px,py,pz,t,d:INT ARRAY w(3)
FOR i=0 TO 2 DO w(i)=v(i)-foc(i) OD
IF vflag=1 THEN px=Vdot(w,M)/128
  py=Vdot(w,M+6)/128
  sx=cx+mag*px/2:sy=cy-mag*py/2
ELSE d=eye(3):t=mag*d/8
  px=Vdot(w,M)/128:py=Vdot(w,M+6)/128
  pz=Vdot(w,M+12)/128
  d==-pz:IF d<4 THEN d=4 FI:d==/4
  sx=t*px/d:sy=t*py/d:sx==+cx:sy=cy-sy
FI RETURN

PROC CLR():Zero(sa1,7680):RETURN

PROC Draw(INT ARRAY P):BYTE i
INT ARRAY pt
pt=P+10
FOR i=1 TO P(0) DO pt==+8 Maksxsy(pt)
  Kolor(pt(3) & 15)
  IF pt(3)<16 THEN LineTo(sx,sy)
  ELSE Dot(sx,sy):xnow=sx:ynow=sy FI
OD RETURN
```

LISTING 5

```
; UPDATES (LISTING 5)

PROC Uview():Position(22,15)
vflag==+1:IF vflag>1 THEN vflag=0 FI
IF vflag=0 THEN Print("perspective")
ELSE Print("orthogonal ") FI:RETURN

PROC Ueye():BYTE i:Position(5,17)
FOR i=0 TO 2 DO
  IF eye(i)<-10 THEN eye(i)=-10
  ELSEIF eye(i)>10 THEN eye(i)=10 FI
  PrintI(eye(i)):Put(' )
OD Position(24,17)
IF eye(3)>200 THEN eye(3)=200
ELSEIF eye(3)<10 THEN eye(3)=10 FI
PrintI(eye(3)):Put(' ):RETURN

PROC Umag():Position(37,17):mag==+1
IF mag>9 THEN mag=1 FI:PrintI(mag):RETURN

PROC Urot():Position(12,18):PrintI(rx)
Put(' ):PrintI(ry):Put(' )
PrintI(rz):Pb(2)
FixR():FixM():CLR():Draw(P):RETURN

PROC Uri():Position(35,18)
PrintI(ri):Put(' ):RETURN

PROC Ucen(INT x,y):Position(8,19)
PrintI(x):Put(' )
PrintI(y):Put(' ):RETURN

PROC Ufoc(INT ARRAY v):BYTE i
Position(24,19)
FOR i=0 TO 2 DO PrintI(v(i)):Put(' ) OD
RETURN

PROC Upt(INT n):Position(5,22):PrintI(n)
Put(' ):IF n<100 THEN Put(' ) FI
Position(30,22):Pb(8):RETURN

PROC Ucxyz(INT ARRAY pt):BYTE i
Position(15,22):PrintB(pt(3) & 3)
IF pt(3)<16 THEN Print("  LineTo ")
ELSE Print("   Plot   ") FI
FOR i=0 TO 2 DO PrintI(pt(i)):Put(' ) OD
RETURN

PROC Ugr():Position(1,15)
IF gflag=7 THEN gflag=8:cx==+80:CP=C8
  Print("xox GRAPHICS 8 xox"):Gr8(dl1)
ELSE gflag=7:cx==-80:CP=C7
  Print("x GRAPHICS 7PLUS x"):Gr7(dl1)
FI Ucen(cx,cy)
CPon():CLR():Draw(P):RETURN

PROC UReset():BYTE i
Position(5,17):Pb(12)
Position(12,18):Pb(12)
Position(8,19):Pb(9)
Position(24,19):Pb(14)
FOR i=1 TO 8 DO P(i)=Q(i) OD
mag=3:rx=0:ry=0:rz=0:ri=30:FixE()
Ueye():Ufoc(foc):Umag():Urot():Uri()
cy=90:cx=160:gflag=8:Ugr():RETURN
```

LISTING 6

```
; STICK3D (LISTING 6)

PROC JoyD(INT n) ;0-EDIT,1-ADD
BYTE i,k,f:INT ARRAY pt
IF n=0 THEN f=0 ELSE f=1:n=P(0) FI
n==+1:pt=P+10+8*n:Upt(n):Ucxyz(pt)
Kolor(pt(3) & 15):Maksxsy(pt):Dit(sx,sy)
DO IF n>200 THEN EXIT FI
  WHILE Strig(0)=0 DO
    st=Stick(0):Dit(sx,sy)
    IF st<15 THEN pt(2)==-jy(st)
      Maksxsy(pt):Ucxyz(pt)
    FI Dit(sx,sy)
  OD
  st=Stick(0):Dit(sx,sy)
  IF key<255 THEN k=key:key=255
    IF k=33 THEN n==+1:Upt(n)
      IF pt(3)<16 THEN LineTo(sx,sy)
      ELSE Dot(sx,sy):xnow=sx:ynow=sy FI
      IF f=1 THEN
        FOR i=0 TO 3 DO pt(4+i)=pt(i) OD
      FI pt==+8:Kolor(pt(3) & 15)
      Maksxsy(pt):Ucxyz(pt)
    ELSEIF k=10 THEN pt(3)==! 16:Ucxyz(pt)
    ELSEIF k=18 THEN IncC()
      pt(3)=(pt(3) & 16)+cnow:Ucxyz(pt)
    ELSE key=k:EXIT FI
  FI
  IF st<15 THEN pt(0)==+jx(st)
    pt(1)==-jy(st):Maksxsy(pt):Ucxyz(pt)
  FI Dit(sx,sy)
OD P(0)=n-1:RETURN

PROC JoyF():BYTE i:INT ARRAY pt(3)
IF cnow=0 THEN Kolor(1) FI
FOR i=0 TO 2 DO pt(i)=foc(i) OD
Maksxsy(pt):Dit(sx,sy):Ufoc(pt)
DO
  WHILE Strig(0)=0 DO
    st=Stick(0):Dit(sx,sy)
    IF st<15 THEN pt(2)==-jy(st)
      Maksxsy(pt):Ufoc(pt)
    FI Dit(sx,sy)
  OD
  st=Stick(0):Dit(sx,sy)
  IF key=33 THEN key=255
    FOR i=0 TO 2 DO foc(i)=pt(i) OD
    CLR():Draw(P):Maksxsy(pt)
  ELSEIF key<255 THEN EXIT FI
  IF st<15 THEN pt(0)==+jx(st)
    pt(1)==-jy(st):Maksxsy(pt):Ufoc(pt)
  FI Dit(sx,sy)
```

```
OD Ufoc(foc):RETURN

PROC JoyE():INT x,y,z
x=eye(0):y=eye(1):z=eye(2)
DO st=Stick(0):Ueye()
  WHILE Strig(0)=0 DO st=Stick(0)
    eye(2)==-jy(st):eye(3)==+jx(st)
    Ueye()
  OD
  IF key=33 THEN key=255
    FixE():FixM():CLR():Draw(P)
    x=eye(0):y=eye(1):z=eye(2)
  ELSEIF key<255 THEN EXIT FI
  eye(0)==+jx(st):eye(1)==-jy(st)
OD eye(0)=x:eye(1)=y:eye(2)=z:Ueye()
RETURN

PROC JoyC():INT x,y
IF cnow=0 THEN Kolor(1) FI
x=cx:y=cy:Dit(x,y)
DO st=Stick(0):Ucen(x,y):Dit(x,y)
  IF key=33 THEN key=255
    cx=x:cy=y:CLR():Draw(P)
  ELSEIF key<255 THEN EXIT FI
  x==+jx(st):y==+jy(st):Dit(x,y)
OD Ucen(cx,cy):RETURN
```

LISTING 7

```
; DISKIO (LISTING 7)

MODULE:BYTE err
BYTE ARRAY fln(16),abort=" ABORTED "

PROC MyErr(BYTE e)
Position(1,16):Print(" ERROR ")
PrintB(e):err=1:mes^=lin21
Position(12,21):PrintB(e)
WHILE key=255 DO OD key=21:RETURN

PROC CIO=$E456(BYTE a,x)

PROC IO2(BYTE cmd CARD buf,len)
BYTE IOcmd=866 ;7-LOAD 11-SAVE
CARD IObuf=868,IOlen=872
IOcmd=cmd:IObuf=buf:IOlen=len
CIO(0,32):RETURN

PROC Dir():BYTE i,j,lft=82
BYTE ARRAY a(18),f(9)
lft=22:Position(22,3)
FOR i=1 TO 11 DO Pb(16):PutE() OD
Position(22,3):j=0
Close(2):Open(2,"D:*.V3D",6,0)
FOR i=1 TO 22 DO InputSD(2,a)
  IF a(0)=16 THEN EXIT FI
  SCopyS(f,a,3,10):Print(f)
  IF j=0 THEN j=1:Put(' )
  ELSE j=0:PutE() FI
OD Position(22,14):Print(a)
Close(2):lft=1:RETURN

PROC Inp(BYTE ARRAY f):BYTE i
BYTE ARRAY a(10)
SCopy(f,"D:            ")
Position(23,16):Pb(15)
Position(23,16):InputS(a)
i=a(0)+3:IF i>11 THEN i=11 FI
SAssign(f,a,3,10):SAssign(f,".V3D",i,14)
Position(23,16):Print(f):RETURN

PROC SaveP():BYTE k:CARD n,t
Position(1,16):Print(" S A V E  ")
Inp(fln)
Position(1,16):Print("[S]- SAVE ")
WHILE key=255 DO OD k=key:key=255
IF k<>62 THEN Position(1,16):Put('S)
  Print(abort):RETURN
FI t=Error:Error=MyErr:err=0
n=8*P(0)+18:Close(2):Open(2,fln,8,0)
IF err>0 THEN Close(2):Error=t:RETURN FI
IO2(11,P,n):IO2(11,C7,5):Close(2)
Position(1,16):Print(" SAVED  ")
Error=t:Dir() RETURN

PROC LoadP():BYTE k:CARD n,t
Position(1,16):Print(" L O A D  ")
Inp(fln)
Position(1,16):Print("[L]- LOAD ")
WHILE key=255 DO OD k=key:key=255
IF k<>0 THEN Position(1,16):Put('L)
  Print(abort):RETURN
FI t=Error:Error=MyErr:err=0
Close(2):Open(2,fln,4,0)
IF err>0 THEN Close(2):Error=t:RETURN FI
IO2(7,P,2):n=8*P(0)+16
IO2(7,P+2,n):IO2(7,C7,5):Close(2)
Position(1,16):Print(" LOADED  ")
Error=t:SetDfault():CPon()
Position(30,19):Pb(6):RETURN
```

LISTING 8

```
; PRINTIO (LISTING 8)

PROC Ppt(INT ARRAY v)
PrintD(2,"("):PrintID(2,v(0))
PrintD(2," "):PrintID(2,v(1))
PrintD(2," "):PrintID(2,v(2))
PrintD(2,":"):PrintID(2,v(3))
PrintD(2,") "):RETURN

PROC Prnt():BYTE i,j,k:CARD n,t
BYTE ARRAY a(13),st=[3 27 51 16]
  ,pre=[4 27 75 192 0],s,d(193)
Position(1,16):Print(" Pic Data  ")
WHILE key=255 DO OD k=key:key=255
IF k<>10 AND k<>58 THEN Position(1,24)
  Put('P):Print(abort):RETURN
FI:t=Error:Error=MyErr:err=0
Close(2):Open(2,"P:",8,0)
IF err>0 THEN Close(2):Error=t:RETURN FI
b(0)=12:SCopy(a,b):SCopyS(a,fln,3,14)
PrintDE(2," "):PrintDE(2,a)
IF err>0 THEN Close(2):Error=t:RETURN FI
IF k=10 THEN st(3)=16:PrintDE(2,st)
  s=sa1:d(0)=192
  FOR i=0 TO 39 DO n=7640+i
    FOR j=1 TO 192 DO d(j)=s(n):n==-40 OD
    PrintD(2,pre):PrintDE(2,d)
  OD
ELSE st(3)=20:PrintDE(2,st)
  PrintD(2,"eye="):Ppt(eye)
  PrintD(2," focus="):Ppt(foc)
  PrintDE(2," "):i=0:j=0:n=P+10
  DO i==+1:IF i>P(0) THEN EXIT FI
    j==+1:n==+8:Ppt(n)
    IF j>2 THEN j=0:PutDE(2) FI
  OD PrintDE(2," ")
FI Position(1,16):Print(" PRINTED  ")
Error=t:Close(2):RETURN
```

LISTING 9

```
; MENU3D (LISTING 9)

PROC Menu():BYTE i,lft=82:BYTE ARRAY dl
dl=dl2-5:dlist=dl:dl2=dlist
```

continued on next page

```
FOR i=0 TO 12 DO dl(i)=dl(5+i) OD
dl(1)=48:dl(2)=48:dl(3)=71:dl(6)=6
dl(8)=48:dl(22)=6:dl(23)=6:dl(24)=16
dl(26)=16:dl(28)=16:dl(30)=16
dl(32)=16:dl(35)=dl&255:lft=1
Print(" v i e w   3 D    ")
PrintE("XOXOXOXOXOXOXOXOXO")
PrintE("        Paul Chabot          1985")
PrintE(" PreRotate  G  Gr7/8      DIRECTORY      ")
PrintE(" X  Inc rx  V  View  ")
PrintE(" Y  Inc ry  M  Mag.  ")
PrintE(" Z  Inc rz  C  Color ")
PrintE(" I  Inc ri     I / O     ")
PrintE(" J  Dec ri  D L S P  ")
PrintE("     Joystick-Trigger    ")
PrintE(" 1 ADD Points/Lines  ")
PrintE(" 2 EDIT Points/Lines ")
PrintE(" 3 Change EYE/dist.  ")
PrintE(" 4 Change FOCUS      ")
PrintE(" 5 Change CENTER     ")
PrintE(" 0 Reset ALL Values  ")
Print("x GRAPHICS 7PLUS x ")
PrintE(" perspective view ")
PrintE("           filename>>")
PrintE("eye                 dist.          mag")
PrintE("PreRotated                          ri=")
PrintE("Center              Focus")
PrintE("   [SPACE] - Picture       [B] - Menu   ")
PrintE("  ERROR         [any key] - MENU")
PrintE("pt#      color")
Print(" C olor    H ue     L um     D efault")
lin15=sa+600
lin16=sa+640:lin17=sa+680:lin18=sa+720
lin19=sa+760:lin20=sa+800:lin21=sa+840
lin22=sa+880:lin23=sa+920:RETURN
```

LISTING 10

```
;  VIEW3D (LISTING 10)
;   by   Paul Chabot
;    ANTIC Magazine

INCLUDE "D:GR78M.ACT"
INCLUDE "D:MISC1.ACT"
INCLUDE "D:COLORS.ACT"
INCLUDE "D:DRAW3D.ACT"
INCLUDE "D:UPDATES.ACT"
INCLUDE "D:STICK3D.ACT"
INCLUDE "D:DISKIO.ACT"
INCLUDE "D:PRINTIO.ACT"
INCLUDE "D:MENU3D.ACT"

PROC Setup():BYTE i
SetTrig():Setjxjy():SetDfault():FixP(0)
Gr78ON():dl1=dlist:sa1=sa:ram==-33
Graphics(0):ram==+33:dl2=dlist:cur=1
CMon():Menu():Dir():UReset():dlist=dl1
mes^=lin20
RETURN

PROC Main():BYTE k:Setup()
DO WHILE key=255 DO OD k=key:key=255
  dlist=dl1:CPon()
  IF k=31 THEN mes^=lin22 ;1
    JoyD(1)
  ELSEIF k=30 THEN mes^=lin22 ;2
    CLR():JoyD(0)
  ELSEIF k=26 THEN mes^=lin17 ;3
    JoyE()
  ELSEIF k=24 THEN mes^=lin19 ;4
    JoyF()
  ELSEIF k=29 THEN mes^=lin19 ;5
    JoyC()
  ELSEIF k=50 THEN mes^=lin20 ;0
    UReset()
  ELSEIF k=22 THEN mes^=lin18 ;X
    rx=(rx+ri+360) MOD 360:Urot()
  ELSEIF k=43 THEN mes^=lin18 ;Y
    ry=(ry+ri+360) MOD 360:Urot()
  ELSEIF k=23 THEN mes^=lin18 ;Z
    rz=(rz+ri+360) MOD 360:Urot()
  ELSEIF k=13 THEN mes^=lin18 ;I
    ri==+1:IF ri>45 THEN ri=45 FI:Uri()
  ELSEIF k=1 THEN mes^=lin18 ;J
    ri==-1:IF ri<-45 THEN ri=-45 FI:Uri()
  ELSEIF k=18 THEN mes^=lin23 ;C
    FixCol()
  ELSEIF k=16 THEN mes^=lin15 ;V
    Uview():CLR():Draw(P)
  ELSEIF k=37 THEN mes^=lin17 ;M
    Umag()
  ELSEIF k=58 THEN dlist=dl2 ;D
    CMon():Dir()
  ELSEIF k=0 THEN mes^=lin16 ;L
    LoadP():Ueye():Ufoc(foc):FixE()
    FixM():CLR():Draw(P):mes^=lin20
  ELSEIF k=62 THEN mes^=lin16 ;S
    SaveP():mes^=lin20
  ELSEIF k=61 THEN mes^=lin15 ;G
    Ugr()
  ELSEIF k=10 THEN mes^=lin16 ;P
    Prnt():mes^=lin20
  ELSEIF k=33 THEN mes^=lin20 ;SPACE
    CLR():Draw(P)
  ELSE dlist=dl2:CMon()
  FI
OD
```

---

enhance your picture details like NASA does!

PIXEL SCANNER

Article on page 14.

LISTING 1

Don't type the TYPO II Codes!

```
WY 10 REM GTIA PIXEL AVERAGING
ON 20 REM BY LYN BUCHANAN
FW 30 REM (c) 1985, ANTIC PUBLISHING
ML 40 GOTO 300
IK 100 REM PIXEL AVERAGING ROUTINE
RM 110 REM SET AREA TO BE ENHANCED
VS 111 REM Y-COORDINATE RANGE
FV 112 FOR Y=1 TO 88
VP 113 REM X-COORDINATE RANGE
EL 114 FOR X=1 TO 77
FS 115 REM STOP ATTRACT MODE
JR 116 POKE 77,0
SX 120 REM LOCATE EACH PIXEL AND ASSIGN I
   TS VALUE TO A VARIABLE
DN 121 LOCATE X-1,Y-1,A
KP 122 LOCATE X-1,Y,B
DZ 123 LOCATE X-1,Y+1,C
JF 124 LOCATE X,Y-1,D
XF 125 LOCATE X,Y,E
JR 126 LOCATE X,Y+1,F
HV 127 LOCATE X+1,Y-1,G
OL 128 LOCATE X+1,Y,H
```

```
IH 129 LOCATE X+1,Y+1,I
CG 130 REM AVERAGE ALL PIXEL VALUES
XG 131 K=INT((A+B+C+D+E+F+G+H+I)/9)
YF 132 REM LIGHTEN THE BACKGROUND
YL 133 IF K=0 THEN K=8
DF 140 REM REPLOT PIXEL IN NEW COLOR
GB 141 COLOR K:PLOT X,Y+95
UA 150 REM RETURN FOR THE NEXT PIXEL
US 151 NEXT X:NEXT Y
DC 160 REM LOCK SCREEN IN GR.9 MODE
GP 161 POKE 77,0:SOUND 0,65,14,14:GOTO 16
   1
SF 199 REM
RM 200 REM DRAW STICK FIGURE BIRD
KK 201 GRAPHICS 9:POKE 712,144
EN 210 REM DRAW TAIL
OY 211 FOR X=1 TO 15:COLOR X:PLOT 15+X,75
   :DRAWTO 3+X/4,71-X/2:NEXT X
IK 220 REM DRAW BODY
SJ 221 FOR X=16 TO 4 STEP -1:COLOR X:PLOT
    15+X/0.6,74-X:DRAWTO 60-X/1.8,74-X
TZ 222 PLOT 15+X/0.6,66+X:DRAWTO 60-X/1.8
   ,66+X:NEXT X
QE 230 REM DRAW WINGS
VP 231 FOR X=1 TO 16:COLOR X:PLOT 4,63:DR
   AWTO 18+X,10-X/2:DRAWTO X*2+20,66-X/4:
   DRAWTO X*2+24,66-X/4
QJ 232 DRAWTO 46+X,20-X/2:DRAWTO 75,45:NE
   XT X
QL 240 REM DRAW HEAD
LM 241 COLOR 11:FOR X=1 TO 22:PLOT 48+X/2
   ,60-X/2:DRAWTO 68-X/6,60-X/2:NEXT X
RW 250 REM DRAW BEAK
US 251 COLOR 15:FOR X=1 TO 5:PLOT 73,67:D
   RAWTO 48+X,67-X:NEXT X:FOR X=1 TO 15:P
   LOT 49+X/2,67:DRAWTO 75,55:NEXT X
KN 260 REM DRAW EYE
RY 261 COLOR 0:FOR Y=54 TO 56:PLOT 60,Y:D
   RAWTO 62,Y:NEXT Y
LV 270 GOTO 100
SG 299 REM
XE 300 GRAPHICS 0:POKE 710,28:POKE 712,28
   :POKE 709,2
QY 301 ? :? :? ,"GTIA PIXEL AVERAGING":?
   ,"  by  Lyn Buchanan":?
NR 302 ? "This demo program draws a stick
   -figurebird using normal PLOT and DRAW
   TO"
ZT 303 ? "commands.  Then the program 'co
   mputer enhances' the image by sampling
    the"
YR 304 ? "color value of each pixel, and"
   :? "averaging it with the values of th
   e   surrounding pixels.":?
SD 305 ? "The BASIC routine is slow (take
   s about12 minutes), but the results ar
   e worththe wait.":?
ZK 310 POKE 752,1:POSITION 12,22:? "READY
   ?  PRESS  START ":POKE 53279,8
VH 320 IF PEEK(53279)<>6 THEN 320
SO 330 POKE 53279,8:GOTO 200
```

your Atari master weaver

COLOR INKLE LOOM

Article on page 29.

LISTING 1

Don't type the TYPO II Codes!

```
GO 10 REM COLOR INKLE LOOM
XJ 20 REM GERALD M. HAGOPIAN
FW 30 REM (c) 1985, ANTIC PUBLISHING
MF 40 DIM ANS$(3),CL$(40),CL1$(40),SB$(1)
   ,SB1$(1),CR$(1),WEAV$(7684)
VY 50 WEAV$(1)="♥":WEAV$(7684)=WEAV$:WEAV
   $(2)=WEAV$:ADX=ADR(WEAV$)
FG 60 LOD=0
PS 70 GOSUB 820:GOTO 980
ST 80 GOSUB 930
RX 90 SCRN=PEEK(88)+PEEK(89)*256
GO 100 X=0:A=1:B=3
OT 110 IF LOD THEN JNK=USR(ADR("hh▟ɢh▟ᴘh▟
   ᴜh▟ᴛh▟ᴏh▟ɴ█♥ᴉᴘʀᴛғᴛᴘ ɪғᴜғᴘᴘ ɪғɢғɴᴘғғᴏ✣ғ◆"
   ),ADX,SCRN,7680)
KW 120 IF LOD THEN POKE 712,ASC(WEAV$(768
   1,7681))
WR 130 IF LOD THEN FOR Y=708 TO 710:POKE
   Y,ASC(WEAV$(Y+6974,Y+6974)):NEXT Y:GOT
   O 150
WX 140 POKE 712,8:POKE 710,34
BP 150 FOR K=1 TO 79:COLOR K/5:SOUND 0,25
   5-(K*2),10,8:PLOT K,150:DRAWTO K,160:N
   EXT K:SOUND 0,0,0,0
OQ 160 GOSUB 180:POKE 559,34
ME 170 GOTO 200
HN 180 POSITION 0,0:? CHR$(125):POKE 752,
   1:POKE 710,34
JS 190 ? "↑A  B  C  D  E  F  G  H  I  J  K  L
   M  N  O":RETURN
AL 200 REM LOOM SET UP
WY 210 CLOSE #1:OPEN #1,4,0,"K:"
QY 220 IF LOD THEN LOD=0:GOTO 710
XY 230 GOSUB 180:POSITION 0,3:? "Number o
   f threads per harness?":? "(LIMIT 39)
   ";:INPUT H:? :IF H<1 OR H>39 THEN 230
UN 240 GOSUB 180
OG 250 POSITION 0,3:? "SURE? (Y/N)      ":
   GET #1,K:IF K<>89 THEN 230
NA 260 X=1:Y=0:COLOR 8:PLOT X,Y:DRAWTO H,
   Y:PLOT H,Y:COLOR 15:DRAWTO H*2,Y
UR 270 REM HARNESS SET UP
TO 280 CLOSE #1:OPEN #1,4,0,"K:":POKE 752
   ,1
PC 290 X=1:Y=0:ET=4
DC 300 FOR T=1 TO H
NP 310 GOSUB 180:POSITION 0,3:? "Harness
   #1: Thread # ";T:? " Color Selection?"
   :GET #1,K:IF K<64 OR K>79 THEN 310
UK 320 GOSUB 180
OQ 330 CL=K-64
XM 340 CL$(T,T)=CHR$(K)
JT 350 FOR X=X TO X+1:COLOR CL:PLOT X,Y:D
   RAWTO X,ET:NEXT X
FT 360 POSITION 0,3:? "Color OK (Y/N) ?":
   GET #1,K:IF K<>89 THEN X=X-2:IF K<>89
   THEN 310
KN 370 NEXT T
AB 380 X=1:Y=5:ET=Y+4
MT 390 FOR T2=1 TO H
FY 400 GOSUB 180:POSITION 0,3:? "Harness
   #2: Thread # ";T2:? " Color Selection?
   ":GET #1,K:IF K<64 OR K>79 THEN 400
UJ 410 GOSUB 180
MV 420 CL1=K-64
ED 430 CL1$(T2,T2)=CHR$(K)
ID 440 FOR X=X TO X+1:COLOR CL1:PLOT X,Y:
   DRAWTO X,ET:NEXT X
FR 450 POSITION 0,3:? "Color OK (Y/N) ?":
   GET #1,K:IF K<>89 THEN X=X-2:IF K<>89
   THEN 400
FQ 460 NEXT T2
DW 470 POSITION 0,0:? "↖ Press START to w
   eave.":POKE 53279,8
DO 480 IF PEEK(53279)<>6 THEN 480
AB 490 POKE 53279,8
IN 500 REM WEAVING ROUTINE
FI 510 POSITION 0,0:? "↖            WEAV
   ING"
DH 520 FOR K=1 TO 79:COLOR 0:SOUND 0,255-
   (K*2),10,8:PLOT K,150:DRAWTO K,160:NEX
   T K:SOUND 0,0,0,0
OT 530 X=1:Y=0:ET=4
GG 540 FOR R=1 TO 15:X=1
ZT 550 FOR I=1 TO H
DZ 560 SB$=CL$(I,I)
EJ 570 CL=ASC(SB$)-64
WX 580 FOR X=X TO X+1:SOUND 0,RND(1)*255,
   10,8:COLOR CL:PLOT X,Y:DRAWTO X,ET:NEX
   T X
UJ 590 SOUND 0,0,0,0:NEXT I
PP 600 X=1:Y=ET+1:ET=Y+4
ZM 610 FOR I=1 TO H
KG 620 SB$=CL1$(I,I)
RY 630 CL1=ASC(SB$)-64
PH 640 FOR X=X TO X+1:SOUND 1,RND(1)*255,
   10,8:COLOR CL1:PLOT X,Y:DRAWTO X,ET:NE
   XT X
YA 650 SOUND 1,0,0,0:NEXT I:Y=ET+1:ET=Y+4
JU 660 NEXT R
EZ 670 JNK=USR(ADR("hh▟ɢh▟ᴘh▟ᴜh▟ᴛh▟ᴏh▟ɴ█♥
   ᴉᴘʀᴛғᴛᴘ ɪғᴜғᴘᴘ ɪғɢғɴᴘғғᴏ✣ғ◆"),SCRN,ADX,7
   680)
QI 680 WEAV$(7681,7681)=CHR$(PEEK(712))
WJ 690 FOR X=7682 TO 7684:WEAV$(X,X)=CHR$
   (PEEK(X-6974)):NEXT X
TW 700 REM VALUE CHANGE OF WEAVING
ZV 710 ? "↖Do you wish to change the valu
   e? (Y/N) ":GET #1,K:IF K<>89 THEN 780
NP 720 TRAP 730
ZD 730 ? "↖Type a letter between A (darke
   st) and O (lightest)":GET #1,K:IF K<65
   OR K>79 THEN 730
PY 740 V=K-64
UM 750 A=PEEK(712):POKE 712,V:? "↖Value O
   K? (Y/N)";:GET #1,K:IF K<>89 THEN POKE
   712,A
PV 760 IF K<>89 THEN 730
JK 770 TRAP 40000
WB 780 ? "↖Weave again? (Y/N)":GET #1,K:I
   F K=89 THEN GOSUB 1530:IF K=89 THEN 10
   0
CQ 790 ? "↖Return to menu? (Y/N)":GET #1,
   K:IF K=89 THEN POP :GOTO 1010
BM 800 ? "↖ Leave program? (Y/N)":GET #1,
   K:IF K<>89 THEN 780
PM 810 ? "↖          Sure? (Y/N)":GET #1,K
   :IF K=89 THEN GRAPHICS 0:END
MZ 820 REM TITLE ROUTINE
PX 830 GRAPHICS 2+16:COLOR 2:POSITION 0,0
JU 840 FOR LG=0 TO 4:? #6;"▣*▣*▣*▣*▣*▣*
   ▣*▣*▣":FOR DL=1 TO 30:SOUND 0,RND(1)*2
   00,10,8:NEXT DL
VC 850 ? #6;"*▣*▣*▣*▣*▣*▣*▣*▣*▣*":FOR DL=
   1 TO 30:SOUND 1,RND(1)*200,10,8:NEXT D
   L:NEXT LG
HW 860 FOR DL=1 TO 300:NEXT DL:SOUND 0,0,
   0,0:SOUND 1,0,0,0:SOUND 1,96,10,8:POSI
   TION 0,4:? #6;"         THE        "
ZS 870 FOR DL=1 TO 100:NEXT DL:SOUND 0,12
   1,10,8:POSITION 0,5:? #6;"    COLOR  L
   OOM    "
GS 880 POSITION 0,0
LU 890 FOR DL=1 TO 100:NEXT DL:SOUND 2,81
   ,10,8:POSITION 0,6:? #6;"       BY GMH
   "
WV 900 FOR L=1 TO 800:NEXT L
RD 910 FOR S=0 TO 3:SOUND S,0,0,0:NEXT S
ZJ 920 RETURN
WZ 930 REM SET UP TEXT WINDOW
```

```
YT 940 POKE 54286,0:GRAPHICS 8:POKE 87,11
   :POKE 623,192:POKE PEEK(560)+256*PEEK(
   561)+166,143:RESTORE
RP 950 ADDRES=1536:POKE 54286,64:FOR K=0
   TO 18:READ P:POKE ADDRES+K,P:NEXT K:P=
   INT(ADDRES/256):POKE 513,P
VX 960 POKE 512,ADDRES-256*P:POKE 54286,1
   92:DATA 72,173,111,2,41,63,141,10,212,
   141,27,208,169,0,141,26,208,104,64
ZT 970 RETURN
WR 980 REM MENU
KG 990 GRAPHICS 2+16:POSITION 1,5:? #6;"
    *■*■ MENU ■*■*"
QP 1000 FOR D=1 TO 500:NEXT D
VB 1010 GRAPHICS 0:POKE 752,1:POKE 710,21
   0
KZ 1020 POSITION 0,8:? "                   MENU
   ":?
NL 1030 ? "          1 CREATE NEW WEAVING"
AN 1040 ? "          2 SAVE WEAVING"
KZ 1050 ? "          3 LOAD OLD WEAVING"
KF 1060 ? "          4 LEAVE PROGRAM"
KI 1070 ? :? "           Selection ?"
II 1080 CLOSE #1:OPEN #1,4,0,"K:"
MN 1090 GET #1,K:IF K<49 OR K>52 THEN 101
   0
IN 1100 A=K-48
XY 1110 ON A GOSUB 80,1130,1280,810
NJ 1120 GOTO 1010
ZY 1130 REM SAVE PATTERN
HD 1140 ? :? :? " Save to: 1..CASSETTE
                           2..DISKETTE"
PS 1150 GET #1,K:IF K<49 OR K>50 THEN 114
   0
ZO 1160 ON K-48 GOTO 1170,1200
IM 1170 ? :? "Place tape in recorder":? "
   Press RECORD and PLAY":? "Then type ■R
   ETURN■"
HO 1180 GET #1,K
KX 1190 OPEN #2,8,128,"C:":GOTO 1210
RN 1200 ? :? :? "Insert diskette and pres
   s ■RETURN■":GET #1,K:OPEN #2,8,0,"D:WE
   AVING"
VW 1210 ? :? :? :? "     ■SAVING■PATTERN■"
LG 1220 ADHI=INT(ADX/256):ADLO=ADX-ADHI*2
   56
EY 1230 MODE=PEEK(87)
LQ 1240 IO=864:POKE IO+2,11:POKE IO+4,ADL
   O:POKE IO+5,ADHI
UI 1250 POKE IO+8,4:POKE IO+9,30
MK 1260 JNK=USR(ADR("hhh■LV■"),32):CLOSE
   #2
AY 1270 RETURN
BZ 1280 REM LOAD OLD PATTERN
RM 1290 ? :? :? "Load from:  1..CASSETTE
                             2..DISKETTE"
MH 1300 GET #1,K:IF K<49 OR K>50 THEN 130
   0
OS 1310 POKE 752,1
FN 1320 ON K-48 GOTO 1330,1370
JX 1330 ? :? :? "Insert and queue tape"
LC 1340 ? "press ■PLAY■, type ■RETURN■"
HJ 1350 GET #1,K
ME 1360 OPEN #2,4,128,"C:":GOTO 1420
WR 1370 ? :? :? "Insert diskette and pres
   s ■RETURN■"
GV 1380 POKE 764,255
GC 1390 IF PEEK(764)<>12 THEN 1390
FZ 1400 POKE 764,255
CX 1410 OPEN #2,4,0,"D:WEAVING"
DX 1420 ? :? :? :? "     ■LOADING■PATTERN■"

AN 1430 GOSUB 930
LQ 1440 ADHI=INT(ADX/256):ADLO=ADX-ADHI*2
   56
BO 1450 IO=864:POKE IO+2,7:POKE IO+4,ADLO
   :POKE IO+5,ADHI
UP 1460 POKE IO+8,4:POKE IO+9,30
MR 1470 JNK=USR(ADR("hhh■LV■"),32):CLOSE
   #2
YX 1480 POKE 712,ASC(WEAV$(7681,7681))
IE 1490 FOR X=708 TO 710:POKE X,ASC(WEAV$
   (X+6974,X+6974)):NEXT X:POKE 559,0
MT 1500 LOD=1:POP :GOTO 90
ET 1510 REM ERASE WEAVING AND RESET
PU 1520 ? "↖      ERASING OLD PATTERN"
IT 1530 FOR U=1 TO 40:CL$(U,U)="e":CL1$(U
   ,U)="e":NEXT U
CX 1540 SB$=" ":SB1$=" ":CR$=" "
BW 1550 X=1:Y=0:ET=4
DM 1560 FOR R=1 TO 15:X=1
LR 1570 FOR I=1 TO H
OB 1580 SB$=CL$(I,I)
TW 1590 CL=ASC(SB$)-64
KF 1600 FOR X=X TO X+1:SOUND 0,RND(1)*255
   ,10,8:COLOR CL:PLOT X,Y:DRAWTO X,ET:NE
   XT X
YB 1610 SOUND 0,0,0,0:NEXT I
PH 1620 X=1:Y=ET+1:ET=Y+4
LH 1630 FOR I=1 TO H
WC 1640 SB$=CL1$(I,I)
JF 1650 CL1=ASC(SB$)-64
FM 1660 FOR X=X TO X+1:SOUND 1,RND(1)*255
   ,10,8:COLOR CL1:PLOT X,Y:DRAWTO X,ET:N
   EXT X
LR 1670 SOUND 1,0,0,0:NEXT I:Y=ET+1:ET=Y+
   4
JU 1680 NEXT R
BM 1690 RETURN
```

type-in music construction software

THE MUSICIAN

Article on page 37.

LISTING 1

Don't type the TYPO II Codes!

```
OF 10 REM THE MUSICIAN
XQ 20 REM BY ANGELO GIAMBRA
FW 30 REM (c) 1985, ANTIC PUBLISHING
RB 40 SI=1:POKE 65,0:GOTO 2060
DC 50 IF PEEK(20)<TEMPO THEN 50
EM 60 A=PEEK(764):RETURN
YX 70 FOR I=1 TO SI-1:PITCH=ASC(SONG$(I,I
   ))
YC 80 IF PITCH=255 THEN POKE C1,ASC(SONG$
   (I+1,I+1)):POKE C2,ASC(SONG$(I+2,I+2))

JV 90 IF PITCH=255 THEN POKE C3,ASC(SONG$
   (I+3,I+3)):I=I+3:NEXT I:GOTO 200
LH 100 IF PITCH=254 THEN POKE C1,0:POKE C
   2,0:POKE C3,0:NEXT I
JT 110 IF PITCH>128 THEN SUST=1:PITCH=PIT
   CH-128:IF PITCH=125 THEN PITCH=128
QM 120 POKE C0,PITCH:POKE 20,0:IF PITCH=0
    THEN 170
RJ 130 IF X>15 THEN X=5:G=1:GOSUB 210
HG 140 YPOS=ASC(YPOS$(I,I)):IF YPOS>131 AN
   D YPOS<147 THEN YPOS=YPOS-128:POSITION
    X,YPOS:? #6;"♣":X=X+1
QS 150 IF YPOS>203 AND YPOS<219 THEN YPOS
   =YPOS-200:POSITION X,YPOS:? #6;"b":X=X
   +1
XR 160 POSITION X,YPOS:? #6;NT$:POSITION
   X,YPOS-1:? #6;"|":POSITION X,YPOS-2:?
```

continued on next page

```
   #6;" |":X=X+2
DG 170 TEMPO=ASC(T$(I,I))*TEMP:GOSUB 50:IF SUST OR PITCH=0 THEN SUST=0:NEXT I:GOTO 200
UT 180 IF A<>255 THEN POKE 764,255:IF A=28 THEN I=SI
XN 190 POKE 53760,0:NEXT I
LS 200 G=0:POKE C1,0:POKE C2,0:POKE C3,0:POKE C0,0:SUST=0:RETURN
GK 210 POKE 53277,0:NT$="●":X=3:SETCOLOR 4,0,0:C=1:SETCOLOR 0,7,4
TU 220 SETCOLOR 1,1,4:SETCOLOR 2,54,4:GRAPHICS 17:POKE 559,46:POKE 53277,3
PA 230 POKE 16,112:POKE 53774,112
TR 240 POKE 756,CHRBASE/256:POSITION 7,0:? #6;"play"
SK 250 POKE 53761,170:POKE 53763,168:POKE 53765,168:POKE 53767,168:POKE 53768,0:POKE 53775,3:IF G THEN RETURN
FY 260 POKE 53254,60:POKE 53255,188:POKE 53260,0:POKE 54279,PMBASE:FOR L=704 TO 707:POKE L,68:NEXT L
LN 270 FOR L=53256 TO 53259:POKE L,3:NEXT L:POKE 53248,60:POKE 53249,92:POKE 53250,124:POKE 53251,156:GOTO 70
JA 280 INPUT #1,TEMP:INPUT #1,SI
LV 290 FOR I=1 TO SI:INPUT #1,CHA$:SONG$(I,I)=CHA$:NEXT I
QE 300 FOR I=1 TO SI:INPUT #1,CHA$:T$(I,I)=CHA$:NEXT I:FOR I=1 TO SI
QS 310 INPUT #1,CHA$:YPOS(I,I)=CHA$:NEXT I:SI=SI+1:POKE 53762,0:POKE 53764,0:POKE 53768,0:RETURN
RL 320 POSITION 4,22:? #6;"   ";DESC$;"            ":NTFLAG=0:RETURN
IL 330 IF MOVEFLAG THEN POSITION XPOS-1,YPOS:GOTO 360
XS 340 IF SUST THEN POSITION XPOS,YPOS+2:? #6;" "
YO 350 POSITION XPOS,YPOS-2:? #6;" ":POSITION XPOS,YPOS
UZ 360 IF SHARPFLAT=3 THEN ? #6;"/"
FP 370 IF SHARPFLAT=1 THEN ? #6;"♯"
AD 380 IF SHARPFLAT=2 THEN ? #6;"b"
OQ 390 IF MOVEFLAG THEN RETURN
FS 400 MOVEFLAG=1:XPOS=XPOS+1:RETURN
WP 410 IF SHARPFLAT THEN GOSUB 330
LI 420 POSITION XPOS,YPOS+1:? #6;" ":POSITION XPOS,YPOS-3
MP 430 ? #6;" ":POSITION XPOS+1,YPOS+1:? #6;" ":POSITION XPOS+1,YPOS-3:? #6;" "
DO 440 IF SHARPFLAT THEN POSITION XPOS-1,YPOS+1:? #6;" ":POSITION XPOS-1,YPOS-1:? #6;" "
ZG 450 IF SUST THEN POSITION XPOS,YPOS+3:? #6;" "
UF 460 POSITION XPOS,YPOS:? #6;NT$;" "
ZC 470 IF NTTYPE<>1 AND RFLAG=0 THEN POSITION XPOS,YPOS-1:? #6;STEM1$;" ":POSITION XPOS,YPOS-2:? #6;STEM2$;" "
YG 480 IF NTTYPE=8 OR NTTYPE=6 OR NTTYPE=3 THEN POSITION XPOS+1,YPOS-2:? #6;STEMFLAG$
FK 490 IF NTTYPE=6 OR NTTYPE=3 THEN POSITION XPOS+1,YPOS-1:? #6;STEMFLAG$
OL 500 IF NTTYPE=3 THEN POSITION XPOS+1,YPOS:? #6;STEMFLAG$
MC 510 IF DOT=1 THEN POSITION XPOS+1,YPOS:? #6;"■"
LH 520 IF SUST THEN POSITION XPOS,YPOS+2:? #6;"s"
FA 530 IF RFLAG OR NTTYPE=1 THEN POSITION XPOS,YPOS-1:? #6;" ":POSITION XPOS,YPOS-2:? #6;"  "
NI 540 GOTO 320
OG 550 REM POLL JOYSTICK
GI 560 A=STICK(0):IF A=15 OR RFLAG THEN 590
OD 570 IF A=14 THEN IF YPOS>4 THEN YPOS=YPOS-1:GOSUB 410:GOTO 550
YQ 580 IF A=13 THEN IF YPOS<18 THEN YPOS=YPOS+1:GOSUB 410:GOTO 550
XW 590 A=PEEK(764):IF A<>255 THEN POKE 764,255:GOTO 620
RM 600 IF STRIG(0)=0 THEN 1020
PE 610 GOTO 550
OI 620 IF A=42 THEN POSITION 4,22:? #6;"exit  y or n":GOSUB 2490:GOTO 550
EF 630 IF A=46 THEN NT$=CHR$(239):NTTYPE=1:DESC$="whole":RFLAG=0:GOTO 900
WX 640 IF A=8 THEN SONG$(SI,SI)=CHR$(254):SI=SI+1:POSITION 4,22:? #6;"chord off    ":GOTO 550
FI 650 IF A=18 THEN 1780
MU 660 IF A=45 THEN GOSUB 1670:GOTO 550
IV 670 IF A=63 THEN GOSUB 1900:GOTO 1970
JA 680 IF A=0 THEN GOSUB 1900:GOTO 2010
FE 690 IF A=57 THEN NT$=CHR$(239):NTTYPE=2:DESC$="half":RFLAG=0:GOTO 870
NE 700 IF A=40 THEN 940
JN 710 IF A=47 THEN NT$=CHR$(148):NTTYPE=4:DESC$="quarter":RFLAG=0:GOTO 870
BQ 720 IF A=53 THEN NTTYPE=8:STEMFLAG$=CHR$(224):DESC$="eighth":EIGHTFLAG=1:RFLAG=0:NT$=CHR$(148):GOTO 900
GN 730 IF A=27 THEN NTTYPE=6:STEMFLAG$=CHR$(224):DESC$="sixteenth":RFLAG=0:EIGHTFLAG=1:NT$=CHR$(148):GOTO 900
TQ 740 IF A=26 THEN NTTYPE=3:STEMFLAG$=CHR$(224):DESC$="thirty sec":RFLAG=0:EIGHTFLAG=1:NT$=CHR$(148):GOTO 900
CD 750 IF A=34 THEN DOT=1:GOTO 900
GK 760 IF A=7 THEN GOSUB 1270:GOTO 550
GB 770 IF A=62 THEN SUST=1:GOTO 900
JX 780 IF A=54 THEN SI=1:FOR I=1 TO 3:SOUND 0,15,10,10:FOR D=1 TO 8:NEXT D:SOUND 0,0,0,0:NEXT I
UF 790 POSITION 4,22:? #6;"song cleared":GOTO 810
PB 800 GOTO 820
XK 810 FOR I=1 TO 300:NEXT I:FIRSTTIME=0:GOTO 2120
MF 820 IF A=21 THEN SHARPFLAT=2:GOTO 900
NM 830 IF A=90 THEN SHARPFLAT=1:GOTO 900
KO 840 IF A=10 THEN GOSUB 210:FIRSTTIME=0:GOTO 2120
NC 850 IF A=7 THEN GOSUB 550:GOTO 900
PR 860 IF A=35 THEN SHARPFLAT=3:GOTO 900
OX 870 IF EIGHTFLAG THEN POSITION XPOS+1,YPOS-2:? #6;" ":POSITION XPOS+1,YPOS-1:? #6;" "
CN 880 IF  NOT DOT THEN POSITION XPOS+1,YPOS:? #6;" "
HE 890 GOSUB 410:GOTO 550
LQ 900 IF NTTYPE<>1 THEN GOSUB 410:GOTO 550
FN 910 POSITION XPOS,YPOS-2:? #6;" ":POSITION XPOS,YPOS-1:? #6;" ":IF EIGHTFLAG THEN POSITION XPOS+1,YPOS-2:? #6;" "
DQ 920 IF EIGHTFLAG THEN POSITION XPOS+1,YPOS-1:? #6;" ":IF  NOT DOT THEN POSITION XPOS+1,YPOS:? #6;" "
GT 930 GOSUB 410:GOTO 550
CS 940 IF NTTYPE=4 THEN NT$="◣"
FH 950 IF NTTYPE=8 THEN NT$="◢":NTTYPE=9
ND 960 IF NTTYPE=2 THEN NT$="▬"
MC 970 RFLAG=1:IF YPOS=12 THEN GOSUB 410:GOTO 550
UI 980 IF YPOS>11 THEN 1010
FN 990 IF YPOS<11 THEN FOR YPOS=YPOS TO 11 STEP 1:GOSUB 410:NEXT YPOS
QS 1000 GOTO 550
XU 1010 FOR YPOS=YPOS TO 13 STEP -1:GOSUB 410:NEXT YPOS:GOTO 550
OC 1020 NX=(YPOS=4)*4+(YPOS=5)*7+(YPOS=6)*13+(YPOS=7)*19+(YPOS=8)*25+(YPOS=9)*28+(YPOS=10)*34+(YPOS=11)*40
LI 1030 NX=NX+(YPOS=12)*43+(YPOS=13)*49+(YPOS=14)*55+(YPOS=15)*61+(YPOS=16)*64+(YPOS=17)*70+(YPOS=18)*76
SE 1040 IF SHARPFLAT=1 THEN NX=NX-3
QQ 1050 IF SHARPFLAT=2 THEN NX=NX+3
TH 1060 IF SHARPFLAT=3 THEN SHARPFLAT=0
OW 1070 IF RFLAG THEN SOUND 0,10,8,10:GOTO 1090
NM 1080 SOUND 0,VAL(SCALE$(NX,NX+2)),10,10
MM 1090 FOR I=1 TO 50:NEXT I:SOUND 0,0,0,0:TEMPO=100:IF DOT THEN T$(SI+1,SI+1)="."
BR 1100 TEMPO=120*(NTTYPE=1)+60*(NTTYPE=2)+30*(NTTYPE=4)+15*(NTTYPE=8)+7*(NTTYPE=6)+3*(NTTYPE=3)
```

```
DW 1110 IF DOT THEN TEMPO=TEMPO*1.5
BB 1120 PITCH=VAL(SCALES$(NX,NX+2)):IF SUS
   T THEN PITCH=PITCH+128:IF PITCH=256 TH
   EN PITCH=253
AJ 1130 SONG$(SI)=CHR$(PITCH):T$(SI,SI)=C
   HR$(TEMPO)
LY 1140 IF RFLAG THEN SONG$(SI)=CHR$(0):D
   OT=0:SUST=0:DESC$="quarter":NTTYPE=4
TC 1150 YPOS$(SI,SI)=CHR$(YPOS):IF SHARPFL
   AT=1 THEN YPOS$(SI,SI)=CHR$(ASC(YPOS$(SI
   ,SI))+128)
QS 1160 IF SHARPFLAT=2 THEN YPOS$(SI,SI)=C
   HR$(ASC(YPOS$(SI,SI))+200)
BU 1170 REM MAKE NEW NOTE
JX 1180 SI=SI+1:POSITION 3,0:IF NT$=CHR$(
   148) THEN NT$="•"
DI 1190 IF NT$=CHR$(239) THEN NT$="o"
NO 1200 STEM1$=" |":STEM2$=" |":STEMFLAG$="
   •":IF FIRSTTIME=0 THEN FIRSTTIME=1:GOT
   O 1220
XK 1210 GOSUB 410:TRAP 40000:IF RFLAG THE
   N NT$="•":RFLAG=0
ED 1220 MOVEFLAG=0:SVEXPOS=XPOS:XPOS=XPOS
   +2:IF XPOS>15 THEN SUBFL=1:XPOS=3:GOSU
   B 2130
MN 1230 IF SVEXPOS=1 THEN SVEXPOS=3
XN 1240 STEM1$=CHR$(130):STEM2$=CHR$(130)
   :STEMFLAG$=CHR$(224):IF NT$="•" THEN N
   T$=CHR$(148)
BR 1250 IF NT$="o" THEN NT$=CHR$(239)
PU 1260 DOT=0:SUST=0:GOSUB 410:GOTO 550
AL 1270 POSITION 2,22:? #6;"   enter chor
   d  ":POSITION 2,23:? #6;"then press re
   turn"
DT 1280 FOR I=1 TO 3:SOUND 0,15,10,10:FOR
    D=1 TO 6:NEXT D:SOUND 0,0,0,0:NEXT I
OX 1290 CH$="      ":OPEN #1,4,0,"K:":POKE
    702,0:CI=1:XP=XPOS-1
ID 1300 A=PEEK(764):IF A=255 THEN 1300
GR 1310 GET #1,A:POKE 764,255:IF A=155 TH
   EN 1470
ZM 1320 IF CI=6 THEN 1300
CV 1330 CH$(CI,CI)=CHR$(A)
LD 1340 IF CHR$(A)="6" THEN CH$(CI,CI)="|
   ":SIXTH=1
PW 1350 IF CHR$(A)="7" THEN CH$(CI,CI)="┬
   ":SEVENTH=1
TC 1360 CHA$=CH$(CI,CI)
MU 1370 IF CHA$="+" THEN CHA$=CHR$(19):GO
   ODCHAR=1
MT 1380 IF CHA$="-" THEN CHA$=CHR$(18):GO
   ODCHAR=1
EM 1390 IF CHA$="#" THEN CHA$=CHR$(16):GO
   ODCHAR=1
AN 1400 IF CHA$="*" THEN CLOSE #1:POSITIO
   N XPOS-1,20:? #6;"      ":SECONDPASS=0:
   GOTO 1630
HR 1410 IF CHA$="a" OR CHA$="b" OR CHA$="
   c" OR CHA$="d" OR CHA$="e" OR CHA$="f"
    OR CHA$="g" THEN 1450
YT 1420 IF (CHA$="m" OR CHA$="j") AND SEC
   ONDPASS THEN 1460
PJ 1430 IF (GOODCHAR OR SIXTH OR SEVENTH)
    AND SECONDPASS THEN 1460
OS 1440 GOTO 1300
LP 1450 SECONDPASS=1
BB 1460 POSITION XP,20:? #6;CHA$:GOODCHAR
   =0:CI=CI+1:XP=XP+1:GOTO 1300
XU 1470 CLOSE #1:POKE 702,64:SECONDPASS=0
   :CHA$=CH$(1,1)
NI 1480 NX=112*(CHA$="c")+106*(CHA$="d")+
   100*(CHA$="e")+97*(CHA$="f")+91*(CHA$=
   "g")+85*(CHA$="a")+79*(CHA$="b")
LV 1490 IF CI=2 THEN OS1=12:OS2=21:GOTO 1
   610
CV 1500 OS1=12:OS2=21:CI=2:IF CH$(2,2)="#
   " THEN NX=NX-3:CI=3
TK 1510 IF CH$(2,2)="b" THEN NX=NX+3:CI=3
MM 1520 IF SIXTH THEN SIXTH=0:OS2=27
OH 1530 IF SEVENTH THEN SEVENTH=0:OS2=30
OZ 1540 IF CH$(CI,CI)="+" THEN OS2=24
TX 1550 IF CH$(CI,CI)="d" THEN OS1=9:OS2=
   18
VP 1560 IF CH$(CI,CI)="-" THEN OS1=12:OS2
   =18
RX 1570 IF CH$(CI,CI)="m" THEN IF CH$(CI+
   1,CI+1)=" " THEN OS1=9:OS2=21
PA 1580 IF CH$(CI,CI)="m" THEN IF CH$(CI+
   1,CI+2)="aj" THEN OS1=12:OS2=33
BM 1590 IF CH$(CI,CI)="m" THEN IF CH$(CI+
   1,CI+1)="┬" THEN OS1=9:OS2=30
OE 1600 IF CH$(CI,CI)="m" THEN IF CH$(CI+
   1,CI+1)="| " THEN OS1=9:OS2=27
IO 1610 SONG$(SI,SI)=CHR$(255):SONG$(SI+1
   ,SI+1)=CHR$(VAL(SCALES$(NX,NX+2)))
GT 1620 SONG$(SI+2,SI+2)=CHR$(VAL(SCALES$(
   NX-OS1,NX-OS1+2))):SONG$(SI+3,SI+3)=CH
   R$(VAL(SCALES$(NX-OS2,NX-OS2+2)))
BR 1630 SIXTH=0:SEVENTH=0:SI=SI+4:POSITIO
   N 2,22:? #6;"                ":POSITI
   ON 2,23:? #6;"                "
YW 1640 POSITION 7,22:? #6;DESC$:IF CHA$=
   "*" THEN RETURN
LR 1650 SOUND 0,ASC(SONG$(SI-3,SI-3)),10,
   10:SOUND 1,ASC(SONG$(SI-2,SI-2)),10,10
   :SOUND 2,ASC(SONG$(SI-1,SI-1)),10,10
GT 1660 FOR I=1 TO 20:NEXT I:SOUND 0,0,0,
   0:SOUND 1,0,0,0:SOUND 2,0,0,0:RETURN
YY 1670 FOR I=1 TO 3:SOUND 0,15,10,10:FOR
    Q=1 TO 6:NEXT Q:SOUND 0,0,0,0:NEXT I
CC 1680 POSITION 5,20:? #6;"change tempo"
   :POSITION 5,22:? #6;"              "
MH 1690 POSITION 8,22:? #6;CHR$(243);CHR$
   (236);CHR$(239);CHR$(247);CHR$(229);CH
   R$(242):TEMPO=0.25
LU 1700 A=STICK(0)
KV 1710 IF A=14 THEN POSITION 8,22:? #6;C
   HR$(230);CHR$(225);CHR$(243);CHR$(244)
   ;CHR$(229);CHR$(242)
KO 1720 TEMPO=-0.25:GOTO 1700
XS 1730 IF A=13 THEN 1690
MR 1740 IF STRIG(0)=0 THEN 1760
QX 1750 GOTO 1700
WD 1760 TEMP=TEMP+TEMPO:FOR I=1 TO 3:SOUN
   D 0,15,10,10:FOR Q=1 TO 6:NEXT Q:SOUND
    0,0,0,0:NEXT I
IJ 1770 POSITION 5,20:? #6;"
   ":POSITION 4,22:? #6;"   ";DESC$;"
       ":RETURN
NF 1780 SI=SI-1:FOR I=1 TO 3:SOUND 0,15,1
   0,10:FOR D=1 TO 8:NEXT D
YO 1790 SOUND 0,0,0,0:NEXT I:POSITION 7,2
   2:? #6;"cancelled"
RB 1800 POSITION XPOS,YPOS-2:? #6;"  ":PO
   SITION XPOS,YPOS-1:? #6;"  ":POSITION
   XPOS,YPOS:? #6;"  "
BM 1810 IF SHARPFLAT THEN POSITION XPOS-1
   ,YPOS:? #6;" "
IB 1820 IF SUST THEN POSITION XPOS,YPOS+2
   :? #6;" "
QQ 1830 XPOS=SVEXPOS:IF SI<1 THEN SI=1
UD 1840 IF SI<4 THEN 550
SG 1850 IF ASC(SONG$(SI-3,SI-3))=255 THEN
    SI=SI-4:POSITION 3,22:? #6;"chord can
   celled"
FL 1860 FOR I=1 TO 100:NEXT I:POSITION 3,
   22:? #6;"                 "
SD 1870 GOTO 550
RX 1880 IF T$(SI,SI)="4" OR T$(SI,SI)="6"
    OR T$(SI,SI)="8" OR T$(SI,SI)="3" THE
   N STEMFLAG$="\"
YG 1890 GOSUB 320
LJ 1900 GRAPHICS 18:POKE 53277,0:POKE 559
   ,34:? #6;"  ENTER SONG NAME"
EF 1910 OPEN #1,4,0,"K":FILENAME$="D:":I=
   3:? #6:? #6:? #6;"          ";
DG 1920 POKE 16,112:POKE 53774,112
OE 1930 GET #1,CH:IF CH=155 THEN CLOSE #1
   :RETURN
CY 1940 IF CH=126 THEN POSITION (PEEK(85)
   -1),(PEEK(84)):? #6;" ";:POSITION (PEE
   K(85)-1),(PEEK(84)):I=I-1:GOTO 1930
FN 1950 IF CHR$(CH)<"A" OR CHR$(CH)>"Z" T
   HEN 1930
WR 1960 FILENAME$(I)=CHR$(CH):? #6;CHR$(C
   H+128);:I=I+1:GOTO 1930
FW 1970 IF LEN(FILENAME$)=2 THEN OPEN #1,
   8,0,"C:":FOR I=1 TO 128:PUT #1,CH:NEXT
    I:GOTO 1990
GU 1980 OPEN #1,8,0,FILENAME$
```

continued on next page

```
ZP 1990 ? #1;TEMP:? #1;SI-1:FOR I=1 TO SI
   -1:CHA$=SONG$(I):? #1;CHA$:NEXT I:FOR
   I=1 TO SI-1:CHA$=T$(I)
IK 2000 ? #1;CHA$:NEXT I:FOR I=1 TO SI-1:
   CHA$=YPOS(I):? #1;CHA$:NEXT I:CLOSE #1
   :FIRSTTIME=0:GOTO 2120
IT 2010 FIRSTTIME=0:TRAP 2040
BO 2020 IF LEN(FILENAME$)=2 THEN OPEN #1,
   4,0,"C:":FOR I=1 TO 128:GET #1,CH:NEXT
    I:GOTO 2050
UT 2030 OPEN #1,4,0,FILENAME$:GOTO 2050
NM 2040 CLOSE #1:GOTO 2120
AY 2050 SI=1:NT$="•":GOSUB 280:CLOSE #1:G
   OSUB 210:FIRSTTIME=0:GOTO 2120
LK 2060 REM INITIALIZE SCREEN
IQ 2070 DIM NT$(1),STEM1$(1),STEM2$(1),SO
   NG$(1000),T$(1000),YPOS$(1000),SNAME$(1
   2),FILENAME$(12)
XF 2080 DIM SCALE$(117),STEMFLAG$(1),DESC
   $(15):DESC$="quarter":DIM CH$(5),CHA$(
   1)
PD 2090 SCALE$="0280290310330350370400420
   450470500530570600640680720760810850091
   0961021081141211281361441531621731 82"
YH 2100 SCALE$(100)="193204217230243255":
   GOSUB 2200:TEMP=1
VV 2110 C0=53760:C1=53762:C2=53764:C3=537
   66
GR 2120 NT$="•":STEM1$=" |":STEM2$=" |":NTT
   YPE=4:DESC$="quarter":YPOS=12:XPOS=1
IP 2130 POKE 53277,0:GRAPHICS 17:SETCOLOR
    0,0,0:POKE 756,INT(CHRBASE/256):POSIT
   ION 3,0:? #6;"the musician"
CX 2140 POKE 16,112:POKE 53774,112
MM 2150 POSITION 7,22:? #6;DESC$:POKE 559
   ,46:POKE 53277,3:POKE 53254,60:POKE 53
   255,188:POKE 53260,0:POKE 623,4
CZ 2160 FOR I=CHRBASE TO CHRBASE+7:POKE I
   ,0:NEXT I:FOR L=704 TO 707:POKE L,120:
   NEXT L
KY 2170 FOR L=53256 TO 53259:POKE L,3:NEX
   T L:POKE 53248,60:POKE 53249,92:POKE 5
   3250,124:POKE 53251,156
GE 2180 IF SUBFL THEN SUBFL=0:RETURN
SR 2190 GOTO 1190
PR 2200 GRAPHICS 17:P=57856:PMBASE=INT(((
   PEEK(106)-12)*256)/1024)*1024
SD 2210 CHRBASE=INT((((PEEK(106)-16)*256)
   /1024-1))*1024
CT 2220 POKE 16,112:POKE 53774,112
RB 2230 FOR I=1 TO 15:POSITION 3,I:? #6;"
   the musician":NEXT I:? #6:? #6:? #6;"
         rev.1.0"
WS 2240 DIM E$(76):FOR I=1 TO 76:READ A:E
   $(I,I)=CHR$(A):NEXT I
WN 2250 DATA 104,104,133,193,104,133,192,
   104,133,197,104,133
OH 2260 DATA 196,169,226,133,195,169,0,13
   3,194,160,0,162,0,177,194,145
YF 2270 DATA 192,200,192,0,208,247,232,22
   4,5,240,10,230,193,230
OI 2280 DATA 195,160,0,169,0,240,232,162,
   0,160,0,169,0,145,196,200
IC 2290 DATA 192,0,208,249,232,224,5,240,
   8,160,0,230,197,169,0,240,236,96
IE 2300 X=USR(ADR(E$),CHRBASE,PMBASE):P=5
   7344+176:FOR I=CHRBASE+176 TO CHRBASE+
   183:POKE I,PEEK(P):P=P+1:NEXT I
VH 2310 P=57344+184:FOR I=CHRBASE+184 TO
   CHRBASE+191:POKE I,PEEK(P):P=P+1:NEXT
   I
VX 2320 P=57344+24:FOR I=CHRBASE+128 TO C
   HRBASE+135:POKE I,PEEK(P):P=P+1:NEXT I
   :P=57344+112
FI 2330 FOR I=CHRBASE+96 TO CHRBASE+103:P
   OKE I,PEEK(P):P=P+1:NEXT I:P=57344+480
OF 2340 FOR I=CHRBASE+256 TO CHRBASE+263:
   POKE I,PEEK(P):P=P+1:NEXT I
CB 2350 FOR I=CHRBASE+80 TO CHRBASE+87:RE
   AD A:POKE I,A:NEXT I:FOR I=CHRBASE+64
   TO CHRBASE+71:READ A:POKE I,A:NEXT I
UA 2360 FOR I=CHRBASE+48 TO CHRBASE+55:RE
   AD A:POKE I,A:NEXT I
NM 2370 START=PMBASE:PMBASE=PMBASE/256
MP 2380 FOR I=START+384+49 TO START+384+8
   1:POKE I,240:NEXT I
CD 2390 FOR I=START+512+49 TO START+640-4
   1 STEP 8:POKE I,255:FOR Z=I+1 TO I+7:P
   OKE Z,0:NEXT Z:NEXT I
RP 2400 FOR I=START+640+49 TO START+768-4
   1 STEP 8:POKE I,255:FOR Z=I+1 TO I+7:P
   OKE Z,0:NEXT Z:NEXT I
CP 2410 FOR I=START+768+49 TO START+896-4
   1 STEP 8:POKE I,255:FOR Z=I+1 TO I+7:P
   OKE Z,0:NEXT Z:NEXT I
QX 2420 FOR I=START+896+49 TO START+1024-
   41 STEP 8:POKE I,255:FOR Z=I+1 TO I+7:
   POKE Z,0:NEXT Z:NEXT I
SX 2430 POKE 15,INT(CHRBASE/256):A=CHRBAS
   E-(INT(CHRBASE/256)*256):POKE 14,A
YO 2440 POKE 54279,PMBASE:POKE 559,46:POK
   E 53277,3
WT 2450 POKE 53760,0:RETURN
AE 2460 DATA 48,24,28,48,24,28,48,24
OT 2470 DATA 48,62,4,8,8,16,16,32
SK 2480 DATA 0,36,44,52,36,44,52,4
XO 2490 A=PEEK(764):IF A=255 THEN 2490
HE 2500 POKE 764,255:IF A<>43 THEN POSITI
   ON 4,22:? #6;"                 ":RETURN
UM 2510 FOR I=1 TO 20:NEXT I
YZ 2520 X=USR(58484)
```

let your joystick select program hues!

COLOR PALETTE

Article on page 21.

LISTING 1

Don't type the TYPO II Codes!

```
ZC 10 REM COLOR PALETTE
TP 20 REM BY JOHN W. FELTON
FW 30 REM (c) 1985, ANTIC PUBLISHING
JR 40 REM INITIAL COLORS
HT 50 DATA 40,202,148,70
PE 60 DATA 50,82,252,114,172,244,198,0
BB 70 REM DISPLAY LIST
CV 80 DATA 112,112,64,69,0,0,133,0,130,0,
   5,133,0,130,0,5,133,0,2,0,2,66,0,0,2,2
   ,2,2,2,66,0,0,65
EK 90 REM DISP. LIST INTERRUPTS
TU 100 DATA 72,173,200,2,69,79,37,78,141,
   10,212,141,24,208,173,7,6,69,79,37,78,
   141,23,208,169,50,141,0,2,169,6,141
GR 110 DATA 1,2,104,64
QH 120 DATA 72,173,0,6,69,79,37,78,141,10
   ,212,141,22,208,173,1,6,69,79,37,78,14
   1,23,208,173,2,6,69,79,37,78,141,24
YS 130 DATA 208,173,3,6,69,79,37,78,141,2
   5,208,169,106,141,0,2,169,6,141,1,2,10
   4,64
AS 140 DATA 72,173,200,2,69,79,37,78,141,
   10,212,141,24,208,173,7,6,69,79,37,78,
   141,23,208,169,142,141,0,2,169,6,141
```

```
GZ 150 DATA 1,2,104,64
OD 160 DATA 72,173,4,6,69,79,37,78,141,10
   ,212,141,22,208,173,5,6,69,79,37,78,14
   1,23,208,173,6,6,69,79,37,78,141,24
HZ 170 DATA 208,169,188,141,0,2,169,6,141
   ,1,2,104,64
DX 180 DATA 72,173,200,2,69,79,37,78,141,
   10,212,141,24,208,169,224,141,9,212,17
   3,7,6,69,79,37,78,141,23,208,104,64
QW 190 REM V/B INTERRUPT
AL 200 DATA 169,14,141,0,2,169,6,141,1,2,
   76,95,228
BS 210 DATA 104,160,219,162,6,169,6,32,92
   ,228,96
HG 220 REM SCROLL SUBROUTINE
HK 230 DATA 104,104,141,0,0,104,141,0,0,9
   6
MO 240 DATA GREY,GOLD,ORANGE,RED,PINK,VIO
   LET,PURPLE,BLUE,CYAN,LT BLUE,TURQUOISE

KP 250 DATA BLUE-GRN,GREEN,YLLW-GRN,ORNG-
   GRN,LT ORANGE
PF 260 DIM PM$(1024),SC$(700),C(12),CRS$(
   10),BLANK(10),A$(20),BLK$(10),ROUTINE$
   (10)
WT 270 REM TITLE PAGE
YQ 280 POKE 106,PEEK(740)
EA 290 GRAPHICS 0:SETCOLOR 3,5,6:SETCOLOR
    0,14,10:POKE 710,0:POKE 752,1
PE 300 X=PEEK(560)+PEEK(561)*256+9:POKE X
   ,6:POKE X+1,6
WC 310 POSITION 4,4:? "color palette
    _______________"
FA 320 POSITION 18,10:? "BY"
GO 330 POSITION 13,12:? "JOHN W. FELTON"
CN 340 REM INITIALIZE
OK 350 RESTORE 100:FOR N=1550 TO 1778:REA
   D A:POKE N,A:NEXT N
RP 360 POSITION 11,20:? "(BACK IN A MOMEN
   T)"
EY 370 RESTORE 50:FOR N=1 TO 12:READ A:C(
   N)=A:NEXT N
XJ 380 RESTORE 60:FOR N=0 TO 7:READ A:POK
   E 1536+N,A:NEXT N
MI 390 SC$(1)=" ":SC$(700)=SC$:SC$(2)=SC$

YR 400 PMBASE=256*(PEEK(106)-8)
MF 410 BSCH=INT(PMBASE/256-5):BSCL=0
PO 420 FOR N=96 TO 208:POKE PMBASE+N,PEEK
   (57344+N):NEXT N
EY 430 REM SET UP DISPLAY LIST
QA 440 GRAPHICS 17:POKE 559,0
LU 450 DL=PMBASE-256*6:SCRLOW=DL+22:SCRHI
   GH=DL+23
BH 460 RESTORE 80:FOR N=0 TO 32:READ A:PO
   KE DL+N,A:NEXT N
DR 470 POKE SCRLOW,BSCL:POKE SCRHIGH,BSCH
   :POKE DL+33,PEEK(560):POKE DL+34,PEEK(
   561)
XK 480 POKE DL+4,PEEK(88):POKE DL+5,PEEK(
   89):POKE 560,DL-INT(DL/256)*256:POKE 5
   61,DL/256
RI 490 N=PEEK(88)+PEEK(89)*256+400:HIGH=I
   NT(N/256):LOW=N-HIGH*256:POKE DL+30,LO
   W:POKE DL+31,HIGH
GH 500 POKE 512,14:POKE 513,6:POKE 54286,
   192:POKE 1543,10
EE 510 REM INIT SCROLL SUBROUTINE
XM 520 RESTORE 230:FOR N=1 TO 10:READ A:R
   OUTINE$(N,N)=CHR$(A):NEXT N
GU 530 HIGH=INT(SCRLOW/256):LOW=SCRLOW-HI
   GH*256:ROUTINE$(8,8)=CHR$(LOW):ROUTINE
   $(9,9)=CHR$(HIGH)
YA 540 HIGH=INT(SCRHIGH/256):LOW=SCRHIGH-
   HIGH*256:ROUTINE$(4,4)=CHR$(LOW):ROUTI
   NE$(5,5)=CHR$(HIGH)
CH 550 A=USR(1768)
TR 560 REM SET UP P/M GRAPHICS
SF 570 VVTP=PEEK(134)+PEEK(135)*256
JF 580 STARP=PEEK(140)+PEEK(141)*256
OJ 590 OFFSET=PMBASE-STARP
LC 600 HI=INT(OFFSET/256):LOW=OFFSET-HI*2
   56
HA 610 POKE VVTP+2,LOW:POKE VVTP+3,HI
MN 620 PM$(512)="♥":PM$(640)=PM$(512):PM$
   (513)=PM$(512)
ON 630 POKE 623,1:POKE 704,12:POKE 53256,
   1
KN 640 POKE 54279,INT(PMBASE/256):POKE 53
   277,3:POKE 756,INT(PMBASE/256)
UR 650 PM$(9,33)="UUUUUUUU▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒
   ▒"
HK 660 REM DRAW SCREEN
UI 670 TSCH=PEEK(89):TSCL=PEEK(88):POKE 8
   9,BSCH:POKE 88,BSCL
XN 680 POKE 87,0:POKE 752,1:POSITION 0,0:
   ? SC$
TC 690 RESTORE 240:FOR N=0 TO 15:READ A$:
   POSITION 1,N:? "■   ";A$:POSITION 16-(N
   >9),N:? N:POSITION 39,N:? "■":NEXT N
GW 700 POKE 88,TSCL:POKE 89,TSCH:POKE 87,
   0
LS 710 FOR N=1 TO 31 STEP 10:COLOR 33+(N=
   11)+(N=21)*2+(N=31)*130:FOR M=0 TO 6 S
   TEP 3
PX 720 FOR I=0 TO 8:FOR J=0 TO 1:PLOT N+I
   ,M+J-(M=6 AND N=31)*3:NEXT J:NEXT I
JG 730 NEXT M:NEXT N
ZP 740 POKE 87,0:POKE 752,1
QW 750 FOR C=1 TO 12:GOSUB 1280:NEXT C
CY 760 POSITION 1,9:? #6;"■■■■■■■■HUE■■■■■
   ■■■■■■■LUMINANCE■■■■■■■■■■■";
RR 770 POSITION 1,10:? #6;"■■■■■■■■■■■■■■■
   ■■■■0■2■4■6■8■10■12■14■■■■";
HV 780 POKE 89,BSCH:POKE 88,BSCL
ZI 790 CRS$="♥▒▒▒▒▒▒▒▒♥"
JM 800 BLK$="♥♥♥♥♥♥♥♥♥♥"
SU 810 POKE 559,46
GA 820 LY=520:Y=1:X=1:GOTO 920
PL 830 REM UPPER WINDOW ROUTINE
CI 840 IF STRIG(0)=0 THEN 840
HC 850 IF STRIG(0)=0 THEN GOSUB 1350:GOTO
   980
NZ 860 FOR DELAY=1 TO 15:NEXT DELAY
SO 870 GOSUB 1210
YT 880 IF DX=0 AND DY=0 THEN 850
JW 890 POKE 77,0
UO 900 X=X+DX:Y=Y+DY
TD 910 X=X-(X=5)*4+(X=0)*4:Y=Y-(Y=4)*3+(Y
   =0)*3
TO 920 PM$(LY,LY+9)=BLK$
VD 930 LY=Y*21+509
XV 940 POKE 53248,X*40+22
FA 950 PM$(LY,LY+9)=CRS$
QV 960 GOTO 850
PG 970 REM LOWER WINDOW ROUTINE
UM 980 CLOC=1527+X+Y*4:CLOC=CLOC-(Y=1)*82
   4:IF Y=3 AND X=4 THEN CLOC=712
HM 990 C=X+(Y-1)*4:COLR=INT(C(C)/16):LUM=
   C(C)-(COLR)*16
VI 1000 BSCRN=BSCL+BSCH*256
IX 1010 SCLOC=BSCRN+(COLR>2 AND COLR<13)*
   (COLR-2)*40+(COLR>12)*400
ZT 1020 GOSUB 1240
SE 1030 BX=LUM+19+(LUM>9)*(LUM-8)/2:BY=CO
   LR:POSITION BX,BY:? "X":LBX=BX:LBY=BY
OX 1040 IF STRIG(0)=0 THEN 980
YM 1050 GOSUB 1210
LK 1060 IF STRIG(0)=0 THEN POSITION BX,BY
   :? " ":GOSUB 1350:GOTO 840
EU 1070 BX=BX+DX*2:BX=BX+(BX>27)*DX:BY=BY
   +DY
MB 1080 IF BX=LBX AND BY=LBY THEN 1050
NF 1090 POKE 77,0
QV 1100 IF BX>37 OR BX<19 THEN BX=LBX:DX=
   0
FF 1110 IF BY>15 OR BY<0 THEN BY=LBY:DY=0

YK 1120 POSITION LBX,LBY:? " "
SH 1130 IF BSCRN+BY*40<SCLOC THEN SCLOC=S
   CLOC-40:GOSUB 1240
CQ 1140 IF BSCRN+BY*40>SCLOC+200 THEN SCL
   OC=SCLOC+40:GOSUB 1240
HL 1150 POSITION BX,BY:? "X":LBX=BX:LBY=B
   Y
LB 1160 LUM=LUM+DX*2:CC=LUM+16*BY:POKE CL
   OC,CC:C(C)=CC:GOSUB 1260
GB 1170 POKE 88,TSCL:POKE 89,TSCH:GOSUB 1
   300
ZL 1180 POKE 88,BSCL:POKE 89,BSCH
QE 1190 GOTO 1050
XU 1200 REM READ JOYSTICK SUB
```

continued on next page

```
TX 1210 S=STICK(0):DX=(S>4 AND S<8)-(S>8
   AND S<12):DY=(S=5 OR S=9 OR S=13)-(S=6
    OR S=10 OR S=14)
BW 1220 POKE 16,112:POKE 53774,112:RETURN
GW 1230 REM SCROLL SUB
DB 1240 A=USR(ADR(ROUTINE$),SCLOC):RETURN
PF 1250 REM SET TEXT COLOR SUB
GV 1260 A=PEEK(712)-INT(PEEK(712)/16)*16:
   POKE 1543,10-(A>9)*2:RETURN
OH 1270 REM WRITE HUE/LUM SUB
QH 1280 BY=INT(C(C)/16):LUM=C(C)-BY*16
JN 1290 Y=1+(C>4)+(C>8):X=C-INT((C-1)/4)*
   4
HE 1300 POSITION X*10-9-(BY>9),Y*3-1:? "
   ";BY;",";
ID 1310 POSITION X*10-6,Y*3-1:? LUM;" ";
AW 1320 POSITION X*10-3,Y*3-1:IF C(C)<99
   THEN ? " ";:IF C(C)<10 THEN ? " ";
XH 1330 ? C(C):RETURN
UQ 1340 REM BEEP SUB
DF 1350 SOUND 0,50,14,6:FOR N=0 TO 10:NEX
   T N:SOUND 0,0,0,0:RETURN
```

LISTING 2

```
10 ;    COLOR PALETTE INTERRUPTS
20 ;    JOHN W. FELTON
30 ;    (c) 1985, ANTIC PUBLISHING
40 ;
50 DRKMSK =   $4E      ; ATTRACT MODE RE
DUCED LUMINANCE MASK
60 COLRSH =   $4F      ; ATTRACT MODE CO
LOR SHIFT MASK
70 WSYNC =    $D40A    ; WAIT FOR HORIZO
NTAL SYNCHRONIZATION REGISTER
80 COLPF0 =   $D016    ; COLOR REGISTER
FOR PLAYFIELD ZERO
90 COLPF1 =   $D017    ; COLOR REGISTER
FOR PLAYFIELD ONE
0100 COLPF2 = $D018    ; COLOR REGISTER
FOR PLAYFIELD TWO
0110 COLPF3 = $D019    ; COLOR REGISTER
FOR PLAYFIELD THREE
0120 COLOR4 = $02C8    ; BACKGROUND COLO
R REGISTER SHADOW
0130 VDSLST = $0200    ; DISPLAY LIST IN
TERRUPT VECTOR
0140 SYSVBV = $E45F    ; STAGE ONE VERTI
CAL BLANK ENTRY
0150 SETVBV = $E45C    ; SET VERTICAL BL
ANK VECTOR ROUTINE
0160 CHBASE = $D409    ; CHARACTER BASE
ADDRESS
0170      .ORG $0600   ; POSITION ON PAG
E SIX
0180 COLR0 .DS 1       ; DEFINE STORAGE
FOR DISPLAY COLORS
0190 COLR1 .DS 1
0200 COLR2 .DS 1
0210 COLR3 .DS 1
0220 COLR4 .DS 1
0230 COLR5 .DS 1
0240 COLR6 .DS 1
0250 COLR7 .DS 1
0260       .DS 4       ; EXTRA STORAGE
0270 ;
0280 ; INTERRUPT #1
0290 ;
0300 INT1 PHA          ; PUSH ACCUMULATO
R ON STACK
0310      LDA COLOR4   ; LOAD BACKGROUND
 COLOR
0320      EOR COLRSH   ; PERFORM ATTRACT
 MODE OPERATIONS
0330      AND DRKMSK
0340      STA WSYNC    ; WAIT FOR HORIZO
NTAL BLANK
0350      STA COLPF2   ; STORE IN GR.0 B
ACKGROUND COLOR REGISTER
0360      LDA COLR7    ; LOAD DISPLAY CO
LOR SEVEN
0370      EOR COLRSH   ; PERFORM ATTRACT
 MODE OPERATIONS
0380      AND DRKMSK
0390      STA COLPF1   ; STORE IN GR.0 L
UMINANCE COLOR REGISTER
0400      LDA # <INT2  ; LOAD ADDRESS OF
 NEXT
0410      STA VDSLST   ; INTERRUPT INTO
THE
0420      LDA # >INT2  ; DISPLAY LIST IN
TERRUPT
0430      STA VDSLST+1 ; VECTOR
0440      PLA          ; POP ACCUMULATOR
 OFF OF STACK
0450      RTI          ; RETURN FROM INT
ERRUPT
0460 ;
0470 ; INTERRUPT #2
0480 ;
0490 INT2 PHA          ; PUSH ACCUMULATO
R ON STACK
0500      LDA COLR0    ; LOAD DISPLAY CO
LOR ZERO
0510      EOR COLRSH   ; PERFORM ATTRACT
 MODE OPERATIONS
0520      AND DRKMSK
0530      STA WSYNC    ; WAIT FOR HORIZO
NTAL BLANK
0540      STA COLPF0   ; STORE IN COLOR
REGISTER ZERO
0550      LDA COLR1    ; LOAD DISPLAY CO
LOR ONE
0560      EOR COLRSH   ; PERFORM ATTRACT
 OPERATIONS
0570      AND DRKMSK
0580      STA COLPF1   ; STORE IN COLOR
REGISTER ONE
0590      LDA COLR2    ; LOAD DISPLAY CO
LOR TWO
0600      EOR COLRSH   ; PERFORM ATTRACT
 OPERATIONS
0610      AND DRKMSK
0620      STA COLPF2   ; STORE IN COLOR
REGISTER TWO
0630      LDA COLR3    ; LOAD DISPLAY CO
LOR 3
0640      EOR COLRSH   ; PERFORM ATTRACT
 MODE OPERATIONS
0650      AND DRKMSK
0660      STA COLPF3   ; STORE IN COLOR
REGISTER THREE
0670      LDA # <INT3  ; LOAD ADDRESS OF
 NEXT
0680      STA VDSLST   ; INTERRUPT INTO
0690      LDA # >INT3  ; DISPLAY LIST IN
TERRUPT
0700      STA VDSLST+1 ; VECTOR
0710      PLA          ; POP ACCUMULATOR
 OFF OF STACK
0720      RTI          ; RETURN FROM INT
ERRUPT
0730 ;
0740 ; INTERRUPT #3
0750 ;
0760 INT3 PHA          ; PUSH ACCUMULATO
R ON STACK
```

```
0770      LDA COLOR4   ; LOAD BACKGROUND
 COLOR
0780      EOR COLRSH   ; PERFORM ATTRACT
 MODE OPERATIONS
0790      AND DRKMSK
0800      STA WSYNC    ; WAIT FOR HORIZO
NTAL BLANK
0810      STA COLPF2   ; STORE IN GR.0 B
ACKGROUND COLOR REGISTER
0820      LDA COLR7    ; LOAD DISPLAY CO
LOR SEVEN
0830      EOR COLRSH   ; PERFORM ATTRACT
 MODE OPERATIONS
0840      AND DRKMSK
0850      STA COLPF1   ; STORE IN GR.0 L
UMINANCE COLOR REGISTER
0860      LDA # <INT4 ; LOAD ADDRESS OF
 NEXT
0870      STA VDSLST   ; INTERRUPT INTO
THE
0880      LDA # >INT4 ; DISPLAY LIST IN
TERRUPT
0890      STA VDSLST+1 ; VECTOR
0900      PLA          ; POP ACCUMULATOR
 OFF OF STACK
0910      RTI          ; RETURN FROM INT
ERRUPT
0920 ;
0930 ; INTERRUPT #4
0940 ;
0950 INT4 PHA          ; PUSH ACCUMULATO
R ON STACK
0960      LDA COLR4    ; LOAD DISPLAY CO
LOR FOUR
0970      EOR COLRSH   ; PERFORM ATTRACT
 MODE OPERATIONS
0980      AND DRKMSK
0990      STA WSYNC    ; WAIT FOR HORIZO
NTAL BLANK
1000      STA COLPF0   ; STORE IN COLOR
REGISTER ZERO
1010      LDA COLR5    ; LOAD DISPLAY CO
LOR FIVE
1020      EOR COLRSH   ; PERFORM ATTRACT
 OPERATIONS
1030      AND DRKMSK
1040      STA COLPF1   ; STORE IN COLOR
REGISTER ONE
1050      LDA COLR6    ; LOAD DISPLAY CO
LOR SIX
1060      EOR COLRSH   ; PERFORM ATTRACT
 OPERATIONS
1070      AND DRKMSK
1080      STA COLPF2   ; STORE IN COLOR
REGISTER TWO
1090      LDA # <INT5 ; LOAD ADDRESS OF
 NEXT
1100      STA VDSLST   ; INTERRUPT INTO
1110      LDA # >INT5 ; DISPLAY LIST IN
TERRUPT
1120      STA VDSLST+1 ; VECTOR
1130      PLA          ; POP ACCUMULATOR
 OFF OF STACK
1140      RTI          ; RETURN FROM INT
ERRUPT
1150 ;
1160 ; INTERRUPT #5
1170 ;
1180 INT5 PHA          ; PUSH ACCUMULATO
R ON STACK
1190      LDA COLOR4   ; LOAD BACKGROUND
 COLOR
1200      EOR COLRSH   ; PERFORM ATTRACT
 MODE OPERATIONS
1210      AND DRKMSK
1220      STA WSYNC    ; WAIT FOR HORIZO
NTAL BLANK
1230      STA COLPF2   ; STORE IN GR.0 B
ACKGROUND COLOR REGISTER
1240      LDA #$E0     ; LOAD MSB OF ADD
RESS OF CHARACTER SET IN ROM
1250      STA CHBASE   ; STORE IN CHARAC
TER SET BASE POINTER
1260      LDA COLR7    ; LOAD DISPLAY CO
LOR SEVEN
1270      EOR COLRSH   ; PERFORM ATTRACT
 MODE OPERATIONS
1280      AND DRKMSK
1290      STA COLPF1   ; STORE IN GR.0 L
UMINANCE COLOR REGISTER
1300      PLA          ; POP ACCUMULATOR
 OFF OF STACK
1310      RTI          ; RETURN FROM INT
ERRUPT
1320 ;
1330 ; VERTICAL BLANK INTERRUPT
1340 ;
1350 VBI LDA # <INT1 ; LOAD ADDRESS OF
 FIRST
1360      STA VDSLST   ; INTERRUPT INTO
THE
1370      LDA # >INT1 ; DISPLAY LIST IN
TERRUPT
1380      STA VDSLST+1 ; VECTOR
1390      JMP SYSVBV   ; JUMP TO OS VERT
ICAL BLANK ROUTINE
1400 ;
1410 ; SET VECTOR TO VERTICAL BLANK IN
TERRUPT
1420 ;
1430      PLA          ; DISCARD NUMBER
OF PARAMETERS PASSED IN BASIC CALL
```

continued on next page

```
1440      LDY # <VBI   ; LOAD Y WITH LSB
 OF INTERRUPT ADDRESS
1450      LDX # >VBI   ; LOAD X WITH MSB
 OF INTERRUPT ADDRESS
1460      LDA #6       ; LOAD ACCUMULATO
R WITH SIX FOR IMMEDIATE VBI
1470      JSR SETVBV   ; CALL OS ROUTINE
 TO SET VECTOR
1480      RTS          ; RETURN FROM SUB
ROUTINE
```

fast graphic power from BASIC

G.U.P. THE GREAT

Article on page 45.

LISTING 1

Don't type the TYPO II Codes!

```
KS 10 REM GUP.BAS
MG 20 REM BY DAREK MIHOCKA
FW 30 REM (c) 1985, ANTIC PUBLISHING
MF 40 DIM FN$(20),TEMP$(20),AR$(93)
WP 50 ? "KOutput filename";:INPUT FN$:CAS
   S=0
SB 60 IF FN$(1,1)="D" AND (FN$(2,2)=":" O
   R FN$(3,3)=":") THEN 80
MG 70 TEMP$(1,2)="D:":TEMP$(3)=FN$:FN$=TE
   MP$
MH 80 TRAP 150
FU 90 ? :? :? "Working...please stand by"

LM 100 RESTORE :READ LN:LM=LN:DIM A$(LN):
   C=1
BG 110 AR$="":READ AR$
XS 120 FOR X=1 TO LEN(AR$) STEP 3:POKE 75
   2,255
DC 130 LM=LM-1:POSITION 10,10:? "(Countdo
   wn...T-";INT(LM/10);")      "
QQ 140 A$(C,C)=CHR$(VAL(AR$(X,X+2))):C=C+
   1:NEXT X:GOTO 110
KK 150 NUMHI=INT(LN/256):NUMLO=LN-NUMHI*2
   56
AW 160 OPEN #1,8,0,FN$
UD 170 AD=ADR(A$):ADHI=INT(AD/256):ADLO=A
   D-ADHI*256
RN 180 IO=848:POKE IO+2,11:POKE IO+4,ADLO
   :POKE IO+5,ADHI:POKE IO+8,NUMLO:POKE I
   O+9,NUMHI
EF 190 X=USR(ADR("hhh⊠LV⊡"),16):CLOSE #1
RQ 200 GRAPHICS 0:? "■COMPLETED■"
MG 1000 DATA 1909
XW 1010 DATA 2552550000641040710760680710
   55071039065112065140065030067142064059
   068072069024068042070028071
FK 1020 DATA 0400710960712330722330720000
   00000039050033048040041035051000053052
   041044041052057000000034057
TD 1030 DATA 0000360330500370430000450410
   40047035043033000000000165203041007005
   206170165204041003168189148
XF 1040 DATA 0640372071332200732550571420
   64133205166204224192176034165203160002
   240009074136240005074136240
CZ 1050 DATA 0010741681891050711332141890
   410721332151772140372220005205145214096
   000000255255255255000000000
SS 1060 DATA 000000000000000000000000152400
   15240015240015240063207243252063207243
   252127191223239247251253254
LD 1070 DATA 1652211972032400411760061662
   03134221133203165221045038065133223165
   203045038065197223240069197
PD 1080 DATA 2032400231012091332240320730
   642302031652031972221240050197223240046
   197224208237169000133207032
RY 1090 DATA 0730641652030241012091332030
   760150652302142080002230215165205145214
   165203024101209133203197223
QC 1100 DATA 2082351692551332070320730641
   652031972221240005230203076023065096000
   104104104141110065104104133
KF 1110 DATA 2041041041332211041041332221
   731100651972221144007166221142110065133
   221165204197222144006166222
EM 1120 DATA 1342041332221731100651332030
   32194064230204165204197222208240173110
   065133203032194064096000000
IX 1130 DATA 0000001041412080652382080652
   38208065173212065072169000072173210065
   072169000072173208065072104
BT 1140 DATA 0742402192010012080030760730
   64141208065206208065104104133203104104
   133204133226076182065173210
JF 1150 DATA 0651332031732120651332041332
   26104141211066104133221141210065104104
   133222141212065032214065206
EW 1160 DATA 208065208219096000000000000000
   00000165222197204208003076180064165221
   197203208003076253066176018
KE 1170 DATA 1662031342211332031652041642
   22133222132204132226165221056229203141
   203066072165204133226169000
TR 1180 DATA 1412480701412110661040160061
   41248070078203066238203066165222056229
   204176009072169001141211066
QR 1190 DATA 1040732550241050011412070661
   69000032214066072141207066173205066133
   219169000141207066104024105
KN 1200 DATA 0000322140661732050661332181
   69128133225169230141183066141168066173
   248070240005024102219102218
PG 1210 DATA 1652191412130662010012080051
   65218141213066173211066240020165218073
   255133218165219073255133219
CF 1220 DATA 1691981411830661411680661732
   13066240042032073064024165225101218133
   225165226101219133226165204
CN 1230 DATA 1972262400202302041652041972
   26240012197222240008032073064230204076
   170066032073064165203197221
HI 1240 DATA 2400052302030761490660960000
   00000000000000000000000000160008056
   237203066008046205066014207
GN 1250 DATA 0660420401440062372030660762
   40066109203066136208233176004109203066
   024046205066096165204197222
GN 1260 DATA 1440061642221332221322040320
   73064165204197222240011230204032073064
   165204197222208245096104104
ES 1270 DATA 1041332301041041332311041332
   25133226104133227240236141203066169128
   141207066169000032214066072
FP 1280 DATA 1690001412070661732050661332
   19104032214066173205066133218169255133
   236165225024101218133225165
DD 1290 DATA 2261012191332261700362260481
   81230236165231056229236133204189170069
   141146064169000141207066162
PU 1300 DATA 0080781460641440030241012271
   06110207066202208241234234133229173024
   068240037165230024101229133
DH 1310 DATA 2211652300562292291332030321
   94064165231024101236133204165230056229
   229133203032194064076086067
AU 1320 DATA 1652362080311652311332041652

   30024101229133203032073064165230056229
```

```
   2291332030320730641652291 33
SB 1330 DATA 2280760860671652310241012361
   3320403224506716523105622923613320 4032
   2450670762150671652300 56229
AR 1340 DATA 2281332031652300562292291332
   2103218006416523002410122813322116 5230
   024101229133203032180064096
GC 1350 DATA 0010000000000000000160240160
   2401602400080008000000002401625500 0001
   255255255255255255001001001
GK 1360 DATA 2552550010001041041041410360
   6904101517018902706813320616923414 1141
   067141142067189043068240029
KX 1370 DATA 2012552400202010022400081690
   7414114106707611506816907414114206 7076
   094068169010076096068189038
XG 1380 DATA 0691410700691890540691410380
   6518925007014110806414102606818901 0071
   133209138201012016022173036
EO 1390 DATA 0690321080691730360690410152
   0100904806516900414119100220805817 3036
   069041048141026071173036069
IP 1400 DATA 0410031700721890260690130260
   7103210806910417018903206914103106 9174
   036069160202177214041191205
IU 1410 DATA 0300692080091772140412400130
   3106914521413620823516200818909607 1157
   192002202016247165088024101
XZ 1420 DATA 2081332251411050711650891332
   2614104107216000102416522510907006 9133
   225153105071165226105000133
JL 1430 DATA 2261530410722001921922082310
   9600000000080008000000000400501201 4000000
   040002002001001002002004 0040
BY 1440 DATA 0400400400400400200402552552
   5525224825224825224825425425425525 5248
   252000000104104141033002104
DH 1450 DATA 1410320021041041701692551410
   4600217301121202410102001014101021 2157
   018208173046002208238096072
RU 1460 DATA 1620961690121570660030320862
   2816209616900315706600316916815706 8003
   169069157069003104157075003
HC 1470 DATA 0412400730281570740030320862
   2817304800213321417304900213321516 0007
   177214141030069096083058153
YG 1480 DATA 1531531531531531531531531531
   5315215215215215215215215115115115 1151
   150150150150150149149149148
DA 1490 DATA 1481481471471471461461461451
   4514414414414314314214214114114014 0139
   138138137137136135135134134
AW 1500 DATA 1331321311311301291281281271
   2612512412312312212112011911811711 6115
   114113111110109108107105104
YT 1510 DATA 1031021000099097096094093091 0
   9000880860840820810790760740720700 67065
   062060057053050046043038033
AY 1520 DATA 0270190001041041041332031041
   0413320410413323310413323410410413 3235
   198235160000165234133214165
GE 1530 DATA 2331332151772141322240720410
   9607407407407407416818524407013322 3104
   041159005223133218169000133
VG 1540 DATA 2190240382180382190382180382
   1903821803821916521813321416521902 4105
   224133215160007132232152024
WN 1550 DATA 1012041701652030740740241251
   0507113321218904107210500013321316 5203
   041003170024164232177214133
KQ 1560 DATA 2161690001332171022161022172
   0201624916000017324207004921206921 6145
   212200173242070049212069217
US 1570 DATA 1452121642321360161832302032
   3020323020323020316520320115214401 6056
   233152133203165204024105008
MN 1580 DATA 1332042011841440001642241962
   3524000042000760650700962550000640 00032
   096000000000000000002003002
QD 1590 DATA 0030020030010010010000000030
   0200100100100400800400800400800200 2002
   001001008004000000104104104
VO 1600 DATA 1332031041041332040760730641
   0416200416000010410415314206420020 2208
   247096104162003173010210157
KG 1610 DATA 1420642020162470961650881332
   1216508913321316003918503306414521 2136
   016248169255133207169036141
RY 1620 DATA 1980020960000000000000362061
   502550002240022250020 00064
```

continued on page 76

learn and play guitar chords on your Atari

GUITAR TUTOR

Article on page 35.

LISTING 1

Don't type the TYPO II Codes!

```
RF 1 REM GUITAR TUTOR
TN 2 REM BY FRANK IMBURGIO
FD 3 REM (c) 1985, ANTIC PUBLISHING
HJ 5 CLOSE #1:OPEN #1,4,0,"K:"
OQ 10 DIM TOP$(30),BOT$(30),FRET$(30),STR
   INGS$(30),PATTERN(6),TONES(63):FOR X=0
   TO 63:TONES(X)=32:NEXT X:GOSUB 20
US 15 GOSUB 100:? "↰":? :? :GOSUB 50:GOSU
   B 500:GOSUB 1560:GOTO 15
RP 20 FOR X=1 TO 13:READ A:TOP$(X,X)=CHR$
   (A):NEXT X
RJ 21 DATA 32,32,17,18,23,18,23,18,23,18,
   23,18,5
EV 25 FOR X=1 TO 13:READ A:BOT$(X,X)=CHR$
   (A):NEXT X
TA 26 DATA 32,32,26,18,24,18,24,18,24,18,
   24,18,3
FN 30 FOR X=1 TO 12:READ A:FRET$(X,X)=CHR
   $(A):NEXT X
SG 31 DATA 32,1,18,19,18,19,18,19,18,19,1
   8,4
FC 40 FOR X=1 TO 13:READ A:STRINGS$(X,X)=C
   HR$(A):NEXT X
SL 41 DATA 32,32,124,32,124,32,124,32,124
   ,32,124,32,124
AE 45 RETURN
AC 50 ? TOP$:FOR X=0 TO 6:? X;:? FRET$:?
   STRINGS$:NEXT X:? BOT$
JI 60 POSITION 9,0:GOSUB 700:GOSUB 710
OU 65 POSITION 2,20:? "■■P■■ TO PLAY NUMB
   ERED CHORDS"
QG 70 POSITION 2,21:? "■■S■■ TO SOUND NOT
   ES"
BI 75 POSITION 2,22:? "■ESC■ for new chor
   d.":REM USE INVERSE VIDEO FOR P,S,AND
   ESC
AR 99 RETURN
SQ 100 GOSUB 1500
XN 102 POKE 751,1:POKE 752,1
SN 110 ? "Pick a note to build a chord on
   ."
DY 120 ? "C":? "D":? "E":? "F":? "G":? "A
   ":? "B":? :? "?"
HH 130 GET #1,K:IF K<65 OR K>71 THEN ? "□
   ":GOTO 130
SP 140 TONIC=K:TNC=K
DX 141 GOSUB 1500:? "You have picked ";CH
   R$(K);" as your tonic.":? :? "Now pick
   :"
VQ 142 ? "1)Natural"
```

continued on next page

```
IQ 143 ? "2)Flat"
TW 144 ? "3)Sharp":?
ZU 145 TRAP 145:INPUT ACC:IF ACC<1 OR ACC
   >3 THEN 145
MB 146 TRAP 40000:IF ACC=2 THEN TONIC=TON
   IC+7
VI 147 IF ACC=3 THEN TONIC=TONIC+14
TA 150 GOSUB 1500
GM 160 ? "Pick a chord pattern"
EZ 170 ? "1)Major":? "2)Minor":? "3)7th":
   ? "4)min 7th":?
BD 180 TRAP 180:INPUT CHORD
TT 190 IF CHORD<1 OR CHORD>4 THEN 180
FI 195 CHORD=CHORD-1:TRAP 40000
YY 200 RETURN
EK 500 GOSUB 600:LINE=X:X=X+CHORD
GT 505 POKE 752,1:RESTORE X
GU 510 POSITION 4,2:FOR F=1 TO 6:READ A:?
    CHR$(A);" ";:NEXT F:FRET=4
EJ 515 FOR F=0 TO 5:READ A:PATTERN(F)=A:N
   EXT F
TE 520 FOR F=0 TO 5:FRET=FRET+2:RESTORE P
   ATTERN(F)
CR 530 POSITION 4,FRET
JH 535 FOR I=1 TO 5:READ A:? CHR$(A);"-";
   :NEXT I:READ A:? CHR$(A)
EZ 540 NEXT F
ON 600 IF TONIC=85 THEN TONIC=72
RN 601 IF TONIC=79 THEN TONIC=73
RA 602 IF TONIC=74 THEN TONIC=66
PX 603 IF TONIC=81 THEN TONIC=75
RS 604 IF TONIC=82 THEN TONIC=76
WL 605 IF TONIC=77 THEN TONIC=69
VI 606 IF TONIC=84 THEN TONIC=78
QY 607 IF TONIC=80 THEN TONIC=67
MB 608 IF TONIC=83 THEN TONIC=70
UI 650 X=((TONIC-65)*100)+20000
AV 699 RETURN
OA 700 ? CHR$(K);
HA 701 IF ACC=2 THEN ? "b";
GH 702 IF ACC=3 THEN ? "#";
FK 703 IF CHORD=1 THEN ? "min";
YC 704 IF CHORD=2 THEN ? "7";
PK 705 IF CHORD=3 THEN ? "min7";
OI 706 ? "          ";
ZY 707 RETURN
JQ 710 FOR X=2 TO 16 STEP 2:POSITION 20,X
   +2:? CHR$((X/2)+176):NEXT X:TONE=CHORD

XH 715 I=4:FOR X=4 TO 16 STEP 2
LW 720 POSITION 23,X:? CHR$(TONES(I));
OI 730 ? CHR$(TONES(I+1));
BP 735 CHORD=TONES(I+2):GOSUB 703:I=I+7:N
   EXT X:CHORD=TONE
XP 799 POSITION 23,X:? "■REST■":RETURN :R
   EM USE INVERSE VIDEO FOR REST
NL 1500 ? "K":? :? :? :POKE 752,0:? :POKE
    710,0:RETURN
MW 1555 FOR TIME=1 TO 200:NEXT TIME:RETUR
   N
EZ 1560 TONIC=TNC:GET #1,K
PO 1561 IF CHR$(K)="S" THEN GOSUB 3000
SO 1562 IF K>48 AND K<56 THEN GOSUB 3100
UN 1563 IF K=80 THEN 3200
BB 1565 IF K<>27 THEN 1560
BE 1570 RETURN
LT 2000 DATA 1,19,19,19,19,4
EU 2001 DATA 1,19,19,19,20,4
AP 2002 DATA 1,19,20,20,19,4
TQ 2003 DATA 1,19,20,20,20,4
HK 2004 DATA 20,20,20,20,20,20
FK 2005 DATA 1,19,19,19,20,4
DB 2006 DATA 1,20,20,19,19,4
GA 2007 DATA 20,19,19,19,19,20
HB 2008 DATA 1,19,19,19,19,20
IU 2009 DATA 1,20,19,19,19,4
JF 2010 DATA 20,19,19,19,19,4
ZX 2011 DATA 1,19,19,20,19,20
FB 2012 DATA 1,19,19,19,20,4
GD 2013 DATA 1,19,19,20,19,4
AC 2014 DATA 1,19,20,19,20,4
HJ 2015 DATA 1,19,20,19,19,4
CG 2016 DATA 1,20,19,20,19,4
GT 2017 DATA 1,19,19,20,19,4
AB 2018 DATA 1,19,19,19,20,20
OO 2999 REM ABOVE SETS UP ALL NEEDED
        SHAPES FOR FINGER DOTS.
ZH 3000 REM SOUND NOTES INDIVIDUALLY
GY 3005 RESTORE LINE+99:FOR X=0 TO 4:READ
    A:PATTERN(X)=A:NEXT X
UO 3010 SOUND 0,PATTERN(0),10,8:GOSUB 155
   5
XD 3020 IF CHORD=1 OR CHORD=3 THEN SOUND
   1,PATTERN(4),10,8:GOSUB 1555:GOTO 3040

WC 3030 SOUND 1,PATTERN(1),10,8:GOSUB 155
   5
XN 3040 SOUND 2,PATTERN(2),10,8:GOSUB 155
   5
GS 3050 IF CHORD=2 OR CHORD=3 THEN SOUND
   3,PATTERN(3),10,8:GOTO 3060
TR 3055 SOUND 3,INT((PATTERN(0)/2)-0.5),1
   0,8
SB 3060 GOSUB 1555:GOSUB 1555:FOR X=0 TO
   3:SOUND X,0,0,0:NEXT X:RETURN
CE 3100 REM PUT CHORDS INTO MEMORY
BS 3110 TONE=(K-49)*7:RESTORE LINE+99
CC 3115 READ ONE,THREE,FIVE,SEV,MINTHREE:
   TONES(TONE)=ONE:TONES(TONE+2)=FIVE
CR 3120 IF CHORD=0 OR CHORD=2 THEN TONES(
   TONE+1)=THREE:GOTO 3130
LF 3125 TONES(TONE+1)=MINTHREE
IK 3130 IF CHORD=2 OR CHORD=3 THEN TONES(
   TONE+3)=SEV:GOTO 3140
PQ 3135 TONES(TONE+3)=INT((ONE/2)-0.5)
NM 3140 TONES(TONE+4)=TONIC:TONES(TONE+5)
   =32
OA 3145 IF ACC=2 THEN TONES(TONE+5)=98
BZ 3150 IF ACC=3 THEN TONES(TONE+5)=35
QF 3155 TONES(TONE+6)=CHORD
KE 3160 GOSUB 710:RETURN
XD 3200 REM PLAY CHORDS IN MEMORY
HT 3205 GOSUB 1500:GOSUB 710:POKE 752,1:?
    "■ESC■ TO STOP PLAYING":REM INVERSE
   VIDEO FOR ESC
TG 3210 X=PEEK(764):IF X=28 THEN 3230
HS 3211 IF X=31 THEN K=0
IF 3212 IF X=30 THEN K=1
LN 3213 IF X=26 THEN K=2
LO 3214 IF X=24 THEN K=3
OV 3215 IF X=29 THEN K=4
OW 3216 IF X=27 THEN K=5
OI 3217 IF X=51 THEN K=6
QF 3218 IF X=53 THEN K=7
DQ 3219 IF K<0 OR K>7 THEN 3210
HJ 3220 TONE=K*7:IF TONES(TONE)=32 THEN G
   OSUB 3230:GOTO 3210
QX 3225 FOR X=0 TO 3:SOUND X,(TONES(TONE+
   X)),10,8:NEXT X:GOTO 3210
DM 3230 FOR X=0 TO 3:SOUND X,0,0,0:NEXT X
   :RETURN
EF 19050 REM EXPLAIN THE FOLLOWING DATA :
   FIRST SIX:PRINT OUT Xs AND Os ON TOP :
   i.e.;32 IS SPACE,88 IS X, 79 IS O
DH 19055 REM THE NEXT SIX ARE DATA LINES
   TO READ SHAPES FOR EACH CHORD PATTERN.
   i.e.;2000 IS A FRET W/ NO FINGERS
HZ 19066 REM THE LAST LINES OF DATA IN A
   SERIES   (LINES 20099,20199, etc.)
SC 19068 REM ARE SOUND NUMBERS:TONIC,THIR
   D,SEVENTH,MINOR THIRD.
WX 20000 DATA 88,79,32,32,32,79,2000,2003
   ,2000,2000,2000,2000
WX 20001 DATA 88,79,32,32,32,79,2001,2002
   ,2000,2000,2000,2000
JI 20002 DATA 88,79,32,79,32,79,2000,2014
   ,2000,2000,2000,2000
MG 20003 DATA 88,79,32,79,32,79,2001,2015
   ,2000,2000,2000,2000
HP 20098 REM A - IN THIS ORDER:20000=MAJ:
   20001=MIN:20002=7th:20003=min7.   All
   others (B,Bb, etc) are in same order.
MA 20099 DATA 144,114,96,81,121
LS 20100 DATA 88,32,32,32,32,32,2000,2004
   ,2000,2003,2000,2000
SI 20101 DATA 88,32,32,32,32,32,2000,2004
   ,2005,2002,2000,2000
PT 20102 DATA 88,32,32,32,32,32,2000,2004
   ,2000,2014,2000,2000
AB 20103 DATA 88,32,32,32,32,32,2000,2004
   ,2005,2015,2000,2000
SK 20198 REM B
PE 20199 DATA 128,102,85,72,108
UE 20200 DATA 88,32,32,79,32,79,2001,2015
```

```
   ,2009,2000,2000,2000
UO 20201 DATA 88,32,32,32,32,32,2000,2000
   ,2004,2005,2002,2000
IB 20202 DATA 88,32,32,32,32,79,2001,2015
   ,2016,2000,2000,2000
DB 20203 DATA 88,32,32,32,32,32,2000,2000
   ,2004,2005,2015,2000
SY 20298 REM C
NS 20299 DATA 121,96,81,68,102
WV 20300 DATA 88,88,79,32,32,32,2000,2011
   ,2012,2000,2000,2000
TH 20301 DATA 88,88,79,32,32,32,2008,2017
   ,2005,2000,2000,2000
WB 20302 DATA 88,88,79,32,32,32,2012,2011
   ,2000,2000,2000,2000
TR 20303 DATA 88,88,79,32,32,32,2008,2017
   ,2005,2000,2000,2000
TM 20398 REM D
RY 20399 DATA 108,85,72,61,91
PS 20400 DATA 79,32,32,32,79,79,2017,2006
   ,2000,2000,2000,2000
OY 20401 DATA 79,32,32,79,79,79,2000,2006
   ,2000,2000,2000,2000
CJ 20402 DATA 79,32,79,32,79,79,2017,2009
   ,2000,2000,2000,2000
BP 20403 DATA 79,32,79,79,79,79,2000,2009
   ,2000,2000,2000,2000
UA 20498 REM E
YI 20499 DATA 96,76,64,53,81
OH 20500 DATA 88,88,32,32,32,32,2018,2017
   ,2015,2000,2000,2000
KR 20501 DATA 32,32,32,32,32,32,2004,2000
   ,2006,2000,2000,2000
NJ 20502 DATA 88,88,32,32,32,32,2004,2017
   ,2009,2000,2000,2000
QA 20503 DATA 32,32,32,32,32,32,2004,2000
   ,2009,2000,2000,2000
UO 20598 REM F
RX 20599 DATA 91,72,60,50,76
VU 20600 DATA 32,32,79,79,79,32,2000,2009
   ,2007,2000,2000,2000
OQ 20601 DATA 32,32,32,32,32,32,2000,2000
   ,2004,2000,2006,2000
WJ 20602 DATA 32,32,79,79,79,32,2008,2009
   ,2010,2000,2000,2000
VD 20603 DATA 32,32,32,32,32,32,2000,2000
   ,2004,2000,2009,2000
VC 20698 REM G
VD 20699 DATA 162,128,108,91,136
GT 20700 DATA 32,32,32,32,32,32,2000,2000
   ,2000,2004,2017,2006
QR 20701 DATA 32,32,32,32,32,32,2000,2000
   ,2000,2004,2000,2006
XM 20702 DATA 88,88,32,32,32,32,2003,2008
   ,2000,2000,2000,2000
XT 20703 DATA 32,32,32,32,32,32,2000,2000
   ,2000,2004,2000,2009
LX 20798 REM Ab
GN 20799 DATA 153,121,102,85,120
LE 20800 DATA 88,32,32,32,32,32,2004,2000
   ,2003,2000,2000,2000
RA 20801 DATA 88,32,32,32,32,32,2004,2005
   ,2002,2000,2000,2000
OV 20802 DATA 88,32,32,32,32,32,2004,2000
   ,2014,2000,2000,2000
XZ 20803 DATA 88,32,32,32,32,32,2004,2005
   ,2015,2000,2000,2000
ML 20898 REM Bb
RN 20899 DATA 136,108,91,76,114
EV 20900 REM C FLAT IS REFERRED TO B
   NATURAL
FN 20999 REM C FLAT NOTES ARE B
OJ 21000 DATA 88,32,32,32,32,32,2000,2000
   ,2000,2004,2000,2003
WN 21001 DATA 88,32,32,32,32,32,2000,2000
   ,2000,2004,2005,2002
TE 21002 DATA 88,32,32,32,32,32,2000,2000
   ,2000,2004,2000,2014
FU 21003 DATA 88,32,32,32,32,32,2000,2000
   ,2000,2004,2005,2015
ML 21098 REM Db
YT 21099 DATA 114,91,76,64,96
WN 21100 DATA 88,32,32,32,32,32,2000,2000
   ,2004,2005,2015,2009
PG 21101 DATA 88,88,32,32,32,32,2000,2008
   ,2017,2014,2000,2000
XA 21102 DATA 88,88,32,32,32,32,2004,2005
   ,2011,2000,2000,2000
PL 21103 DATA 88,88,32,32,32,32,2015,2018
   ,2017,2000,2000,2000
MZ 21198 REM Eb
WX 21199 DATA 102,85,68,57,91
PS 21200 REM F FLAT IS REFERRED TO E
JH 21299 REM F FLAT NOTES ARE E
ZL 21300 DATA 32,32,32,32,32,32,2000,2004
   ,2017,2006,2000,2000
ML 21301 DATA 32,32,32,32,32,32,2000,2004
   ,2000,2006,2000,2000
FJ 21302 DATA 32,32,32,32,32,32,2000,2004
   ,2017,2009,2000,2000
SJ 21303 DATA 32,32,32,32,32,32,2000,2004
   ,2000,2009,2000,2000
OB 21398 REM Gb
WG 21399 DATA 173,136,114,96,144
KB 21400 REM A SHARP IS B FLAT
FF 21500 REM B SHARP IS C
MV 21600 REM C SHARP IS D FLAT
OF 21700 REM D SHARP IS E FLAT
JJ 21800 REM E SHARP IS F
QZ 21900 REM F SHARP IS G FLAT
LJ 22000 REM G SHARP IS A FLAT
```

game of the month

HELICOPTER ROUNDUP

Article on page 48.

LISTING 1

Don't type the TYPO II Codes!

```
IU 10 REM HELICOPTER ROUNDUP
ID 20 REM BY WALTER BULAWA
FW 30 REM (c) 1985, ANTIC PUBLISHING
UI 40 GOSUB 1350:REM GAME INITIALIZATION
NJ 50 GOSUB 1160:REM LEVEL INITIALIZATION

BV 60 S=STICK(0)
DT 70 POKE PMBASE+861-FUEL,F*126:FUEL=FUE
   L-0.04:FUELUSED=FUELUSED+1
JS 80 POKE 16,112:POKE 53774,112:IF FUEL<
   0 THEN 340
GS 90 SOUND 0,15,8,10
EC 100 XDIF=H(S-4):YDIF=V(S-4)
EF 110 XPOS=XPOS+4*XDIF:YPOS=YPOS+4*YDIF
LF 120 IF XPOS<40 THEN XPOS=40
TJ 130 IF XPOS>200 THEN XPOS=200
PL 140 IF YPOS<8 THEN YPOS=8
YD 150 IF YPOS>112 THEN YPOS=112
FQ 160 SOUND 0,0,0,0:SOUND 3,0,0,0
DZ 170 IF XDIF=1 THEN PMD=PMR
EN 180 IF XDIF=-1 THEN PMD=PML
```

continued on next page

```
YH 190 A=USR(MOVE,0,PMBASE,PMD+IC,XPOS,YP
OS,5-IC)
SF 200 A=USR(MOVE,3,PMBASE,PMX,XPOS,YPOS,
5)
PA 210 IC=IC+1
BN 220 IF IC=2 THEN IC=0:POKE 232,NCOWS:P
OKE 230,XPOS:POKE 231,YPOS:A=USR(1536,
ADR(CPOS$))
MO 230 IF PEEK(208)=NCOWS THEN IF  NOT DO
NE THEN GOSUB 470:DONE=1
EW 240 F=0
XB 250 IF  NOT (XPOS=176) THEN 60
NL 260 IF  NOT (YPOS=100) THEN 60
JI 270 IF DONE THEN 520
YU 280 SOUND 3,50,14,8
ZU 290 IF STRIG(0) THEN 60
UR 300 POKE PMBASE+861-(FUEL+0.04),126
HN 310 FUEL=FUEL+0.3:IF FUEL>51 THEN FUEL
=51
QD 320 F=1:POKE 77,0:GOTO 60
LS 330 REM END OF MAIN LOOP
BG 340 REM CRASH
EM 350 CRASH=1
CS 360 SOUND 0,100,4,14:SOUND 3,0,0,0
ZQ 370 A=USR(MOVE,0,PMBASE,PMC,XPOS,YPOS,
8)
AH 380 FOR I=20 TO 200 STEP 4
AM 390 IF I=100 THEN POKE 53256,3
KZ 400 SETCOLOR 4,I,8
VB 410 FOR N=1 TO 10:NEXT N
FY 420 NEXT I
CE 430 SETCOLOR 4,0,0:SOUND 0,0,0,0
OO 440 FOR I=1 TO 400:NEXT I
SV 450 POKE 53256,1
OQ 460 GOTO 530
NU 470 REM CLOSE CORRAL GATE
FN 480 COLOR 131:PLOT 6,11:DRAWTO 6,13
KA 490 SOUND 3,40,8,15:FOR I=1 TO 40:NEXT
 I
DO 500 FOR I=150 TO 60 STEP -10:FOR N=I T
O I-50 STEP -5:SOUND 3,N,14,12:NEXT N:
NEXT I:SOUND 3,0,0,0
CQ 510 POSITION 2,21:? #6;"return to pad"
:RETURN
KT 520 SOUND 0,0,0,0:FOR I=1 TO 500:NEXT
I
DI 530 SCORE=( NOT CRASH)*(NCOWS*1000-FUE
LUSED):IF SCORE<0 THEN SCORE=0
AT 540 IF SCORE>HISCORE THEN HISCORE=SCOR
E
JG 550 POKE 53248,0:POKE 53249,0:POKE 532
50,0:POKE 53251,0
OP 560 GRAPHICS 1+16:POKE 756,CHBASE/256
LJ 570 COLOR 131:PLOT 0,0:DRAWTO 19,0:DRA
WTO 19,23:DRAWTO 0,23:DRAWTO 0,0
HS 580 POSITION 2,5:? #6;"HIGH SCORE: ";H
ISCORE
YG 590 POSITION 2,8:? #6;"game score: ";S
CORE
UJ 600 POSITION 3,13:? #6;"fuel used: ";F
UELUSED
OL 610 IF  NOT CRASH THEN POSITION 2,14:?
 #6;"cattle penned: ";NCOWS
VR 620 IF CRASH THEN POSITION 2,16:? #6;"
crash - too bad"
CW 630 POSITION 1,20:? #6;"X PRESS Y STAR
T Z"
ZT 640 POKE 53279,8
YT 650 RESTORE 690
UN 660 READ I:IF I=-1 THEN FOR J=1 TO 200
:NEXT J:GOTO 650
OT 670 IF (PEEK(53279))=6 THEN SOUND 3,0,
0,0:NCOWS=NCOWS+1:GOTO 50
NV 680 SOUND 3,I,10,8:FOR J=1 TO 12:NEXT
J:GOTO 660
IN 690 DATA 60,60,0,60,60,47,47,47,0,47,4
7,0,47,47,47,0,0,0,60,60,0,60,60,45,45
,45,0,45,45,0
HU 700 DATA 45,0,53,53,0,60,60,60,0
YH 710 DATA 60,60,47,47,47,0,47,47,0,47,4
7,53,53,60,60,0,45,45,45,47,0,45,45,0,
60,60,60,0,0,0,-1
AJ 720 REM DRAW PLAYFIELD
BQ 730 GRAPHICS 1+16:POKE 559,0:POKE 756,
CHBASE/256
LD 740 SETCOLOR 0,1,4:SETCOLOR 3,2,8
HY 750 COLOR 131:FOR I=8 TO 17:PLOT 6,I:D
RAWTO 13,I:NEXT I
JE 760 PLOT 0,0:DRAWTO 18,0:DRAWTO 18,23:
DRAWTO 0,23:DRAWTO 0,0
HL 770 POSITION 19,20:? #6;"@":POSITION 1
9,21:? #6;"@":POSITION 19,22:? #6;"@":
POSITION 19,23:? #6;"@";
AX 780 POSITION 15,20:? #6;"QQQ":POSITION
 15,21:? #6;"QWW":POSITION 15,22:? #6;
"QQQ"
TY 790 REM PLOT TREES
VR 800 RESTORE 820
MP 810 FOR I=1 TO 5:READ C:READ R:POSITIO
N C,R:? #6;"v":POSITION C,R+1:? #6;"K"
:NEXT I
JA 820 DATA 8,3,3,5,17,14,4,4,4,18
FN 830 REM PLACE COWS
BF 840 COLOR 89
VM 850 FOR I=1 TO NCOWS
XG 860 COL=INT(17*RND(0))+1:ROW=INT(17*RN
D(0))+1
QP 870 LOCATE COL,ROW,XVAL:IF XVAL<>32 TH
EN GOTO 860
DO 880 POKE ADR(CPOS$)+(I-1)*2,COL:POKE A
DR(CPOS$)+(I-1)*2+1,ROW:PLOT COL,ROW
GQ 890 NEXT I
GC 900 REM PLACE HELICOPTER
OO 910 XPOS=176:YPOS=100:PMD=PML
WA 920 A=USR(MOVE,0,PMBASE,PMD,XPOS,YPOS,
5)
RW 930 POKE 53249,200:POKE 53250,200
RU 940 REM CLEAR CENTER OF CORRAL
KH 950 COLOR 32:FOR I=9 TO 16:PLOT 7,I:DR
AWTO 12,I:NEXT I
QZ 960 PLOT 6,11:DRAWTO 6,13
ZT 970 RETURN
AK 980 REM TITLE PAGE DISPLAY
ZT 990 POKE 106,PEEK(740):POKE 53279,8
ZO 1000 POKE 53248,0:POKE 53249,0:POKE 53
250,0:POKE 53251,0
EV 1010 GRAPHICS 2+16:SETCOLOR 1,1,4
TD 1020 POSITION 5,3:? #6;"HELICOPTER":PO
SITION 6,5:? #6;"ROUNDUP"
XW 1030 COLOR 3:PLOT 0,0:DRAWTO 19,0:DRAW
TO 19,11:DRAWTO 0,11:DRAWTO 0,0
KO 1040 POSITION 4,9:? #6;"PRESS START"
JU 1050 RESTORE 690
GY 1060 READ I:IF I=-1 THEN FOR J=1 TO 20
0:NEXT J:GOTO 1050
GK 1070 IF PEEK(53279)=6 THEN SOUND 3,0,0
,0:RETURN
CE 1080 SOUND 3,I,10,8:FOR J=1 TO 12:NEXT
 J:GOTO 1060
TX 1090 REM PLEAE WAIT DISPLAY
TW 1100 GRAPHICS 2+16
VT 1110 POSITION 7,4:? #6;"PLEASE"
ZY 1120 POSITION 8,7:? #6;"WAIT"
ZA 1130 POSITION 5,10:? #6;"30 SECONDS"
WT 1140 FOR I=1 TO 300:NEXT I
AQ 1150 RETURN
OO 1160 REM LEVEL INITIALIZATION
AL 1170 POKE 53248,0:POKE 53249,0:POKE 53
250,0:POKE 53251,0
FS 1180 GRAPHICS 2+16:SETCOLOR 1,1,4
YM 1190 IC=0:FUEL=51
XR 1200 COLOR 3:PLOT 0,0:DRAWTO 19,0:DRAW
TO 19,11:DRAWTO 0,11:DRAWTO 0,0
QR 1210 POSITION 5,2:? #6;"HELICOPTER":PO
SITION 6,4:? #6;"ROUNDUP"
VU 1220 POSITION 2,7:? #6;"SELECT number 
of"
DF 1230 POSITION 2,8:? #6;"cows to herd: 
";NCOWS
ZE 1240 POSITION 4,10:? #6;"press START"
UE 1250 FOR I=PMBASE+810 TO PMBASE+861:PO
KE I,126:NEXT I
JR 1260 POKE 53279,8
NK 1270 IF PEEK(53279)=5 THEN NCOWS=NCOWS
+1:IF NCOWS>30 THEN NCOWS=1
DF 1280 POSITION 16,8:? #6;NCOWS;" "
UM 1290 FOR I=1 TO 10:NEXT I
VW 1300 IF PEEK(53279)<>6 THEN 1260
HK 1310 SCORE=0
DJ 1320 GOSUB 720:POKE 623,1:POKE 559,46:
FUELUSED=0:CRASH=0:DONE=0
CR 1330 POKE 53256,1:POKE 53259,1
AR 1340 RETURN
VB 1350 REM INITIALIZATION
```

```
QZ 1360 GOSUB 980:GOSUB 1090
MC 1370 HISCORE=0:NCOWS=1
HO 1380 DIM PMMOVE$(100),PR$(5),PL$(5),PC
   $(8),CPOS$(60),CHNEW$(8)
SW 1390 DIM PX$(5)
YQ 1400 DIM H(11),V(11)
QM 1410 MOVE=ADR(PMMOVE$):PMR=ADR(PR$):PM
   L=ADR(PL$):PMC=ADR(PC$)
DA 1420 PMX=ADR(PX$)
EE 1430 RESTORE 1440:FOR I=5 TO 15:READ N
   :H(I-4)=N:READ N:V(I-4)=N:NEXT I
QB 1440 DATA 1,1,1,-1,1,0,0,0,-1,1,-1,-1,
   -1,0,0,0,0,1,0,-1,0,0
HY 1450 GRAPHICS 1+16:POKE 559,0
ZE 1460 PMBASE=INT((PEEK(106)-6)/4)*4
JE 1470 CHBASE=(PMBASE-2)*256
SK 1480 PMBASE=PMBASE*256
OW 1490 PR$(1)=CHR$(31):PR$(2)=CHR$(132):
   PR$(3)=CHR$(206):PR$(4)=CHR$(127):PR$(
   5)=CHR$(14)
IW 1500 PL$(1)=CHR$(248):PL$(2)=CHR$(33):
   PL$(3)=CHR$(115):PL$(4)=CHR$(254):PL$(
   5)=CHR$(112)
SD 1510 PC$(1)=CHR$(90):PC$(2)=CHR$(82):P
   C$(3)=CHR$(34):PC$(4)=CHR$(116):PC$(5)
   =CHR$(92):PC$(6)=CHR$(94)
YA 1520 PC$(7)=CHR$(162):PC$(8)=CHR$(16)
KN 1530 PX$(1)=CHR$(0):PX$(2)=CHR$(0):PX$
   (3)=CHR$(129):PX$(4)=CHR$(0):PX$(5)=CH
   R$(0)
OT 1540 REM ERASE P/M AND CHR MEMORY
UF 1550 RH=PEEK(89):RL=PEEK(88)
BA 1560 I=PEEK(106):POKE 106,I-6:POKE 89,
   PEEK(106)-8:POKE 88,0:? "↖":POKE 89,RH
   :POKE 88,RL:POKE 106,I
RQ 1570 POKE 559,0
NA 1580 REM P/M MOVE ROUTINE
WV 1590 RESTORE 1600:FOR I=1 TO 100:READ
   N:PMMOVE$(I)=CHR$(N):NEXT I
MB 1600 DATA 216,104,104,104,133,213,104,
   24,105,2,133,206,104,133,205,104,133,2
   04,104,133,203,104,104,133,208
BN 1610 DATA 104,104,133,209,104,104,24,1
   01,209,133,207,166,213,240,16,165,205,
   24,105,128,133,205,165,206,105
TN 1620 DATA 0,133,206,202,208,240,160,0,
   162,0,196,209,144,19,196,207,176,15,13
   2,212,138,168,177,203,164
UL 1630 DATA 212,145,205,232,169,0,240,4,
   169,0,145,205,200,192,128,208,224,166,
   213,165,208,157,0,208,96
LG 1640 REM CHAR MOVE ROUTINE
NV 1650 RESTORE 1960:I=0
AO 1660 READ N:IF N=-1 THEN 1680
YO 1670 POKE 1536+I,N:I=I+1:GOTO 1660
HV 1680 CHNEW$="KVQW#XYZ"
RV 1690 FOR I=0 TO 511:POKE CHBASE+I,PEEK
   (57344+I):NEXT I
NT 1700 RESTORE 1740
AR 1710 FOR I=1 TO 8
NV 1720 CHADD=CHBASE+(ASC(CHNEW$(I,I))-32
   )*8
BD 1730 FOR J=0 TO 7:READ N:POKE CHADD+J,
   N:NEXT J:NEXT I
MG 1740 REM TREE TRUNK -K
FA 1750 DATA 24,24,24,24,24,24,24,24
KM 1760 REM TREE LEAVES-V
CS 1770 DATA 64,49,87,221,84,58,120,28
KD 1780 REM LANDING PAD-Q
NA 1790 DATA 255,255,255,255,255,255,255,
   255
RJ 1800 REM REFUEL ZONE-W
WS 1810 DATA 231,231,231,0,0,231,231,231
NW 1820 REM FENCE - #
GC 1830 DATA 255,68,68,68,255,68,68,68
WW 1840 REM LEFT COW - X
AA 1850 DATA 64,192,255,63,63,17,17,17
OF 1860 REM CENTER COW - Y
NK 1870 DATA 36,24,24,60,60,60,36,36
IM 1880 REM RIGHT COW - Z
DB 1890 DATA 2,3,255,252,252,136,136,136
WZ 1900 POKE 54279,INT(PMBASE/256):POKE 5
   3277,3
YP 1910 POKE 704,8:POKE 705,14:POKE 706,1
   86:POKE 707,252
RR 1920 FOR I=PMBASE+640 TO PMBASE+895:PO
   KE I,0:NEXT I
AP 1930 FOR I=PMBASE+682 TO PMBASE+733:PO
   KE I,129:NEXT I:POKE PMBASE+734,255:PO
   KE PMBASE+681,255
BD 1940 RETURN
AD 1950 REM COW MOVE ROUTINE
ER 1960 DATA 104,104,133,204,104,133,203,
   169
KG 1970 DATA 0,133,208,165,232,208,1,96
UU 1980 DATA 198,232,165,232,10,168,177,2
   03
YF 1990 DATA 133,219,200,177,203,133,220,
   32
VA 2000 DATA 145,6,165,230,133,207,165,21
   9
SO 2010 DATA 10,10,10,24,105,48,133,206
NV 2020 DATA 32,193,6,138,133,233,24,101
OL 2030 DATA 219,133,221,32,145,6,165,231
VC 2040 DATA 133,207,165,220,10,10,24,105
IC 2050 DATA 16,133,206,32,193,6,138,24
IA 2060 DATA 101,220,133,222,170,32,165,6
TK 2070 DATA 164,221,177,205,201,0,240,10
DK 2080 DATA 165,220,166,219,32,230,6,76
ES 2090 DATA 11,6,169,57,24,101,233,145
YI 2100 DATA 205,166,220,32,165,6,164,219
TD 2110 DATA 169,0,145,205,165,232,10,168
AU 2120 DATA 165,221,145,203,165,222,200,
   145
WF 2130 DATA 203,166,221,32,230,6,76,11
VU 2140 DATA 6,173,10,210,201,235,176,3
HC 2150 DATA 162,0,96,201,245,176,3,162
CY 2160 DATA 1,96,162,255,96,165,88,133
WU 2170 DATA 205,165,89,133,206,138,208,1
IN 2180 DATA 96,165,205,216,24,105,20,133
AO 2190 DATA 205,144,2,230,206,202,208,24
   1
HM 2200 DATA 96,173,10,210,41,31,133,205
BV 2210 DATA 165,207,24,197,206,176,13,16
   5
HB 2220 DATA 206,56,229,207,24,197,205,17
   6
CK 2230 DATA 2,162,1,96,229,206,24,197
ZM 2240 DATA 205,176,2,162,255,96,24,201
JR 2250 DATA 8,176,1,96,201,17,144,1
KL 2260 DATA 96,138,24,201,7,176,1,96
LU 2270 DATA 201,13,144,1,96,230,208,96
FH 2280 DATA -1
```

logo

TURTLE PIANO

Article on page 10.

LISTING 1

```
TO SIGN :LEN
TELL [0 1] PU
TELL 0 BK 6 TELL 1 FD 14
TELL [0 1] LT 90 FD :LEN / 2 - 10
END

TO STAFF :LEN :HIT
CS PU LT 90 FD :LEN / 2 RT 90 FD :HIT
 * 5 / 2 PD
REPEAT 4 [SETH 90 FD :LEN BK :LEN RT
90 FD :HIT]
RT 180
REPEAT 2 [FD :HIT * 4 RT 90 FD :LEN R
T 90]
SIGN :LEN
END

TO MEASURE
CS
TELL 2 STAFF 250 10
END

TO DIRECTIONS
CT
PR [IF YOU WANT THE COMPUTER TO PLAY]
PR [ALONG WITH YOU, PRESS 'y' NOW.]
MAKE "CHOI RC
IF :CHOI = "Y [PR [THE COMPUTER WILL
PLAY ALONG.]] [PR [THE COMPUTER WILL
NOT PLAY.]]
PR [Press a key to go on.]
PR RC CLEANUP
END

TO SETUP
SETBG 0
TELL 2 SETPN 0 SETPC 0 102 SETC 15
TELL [0 1] SETC 6
TELL [0 1 2] CS
TELL 2 MEASURE
TELL 2 PU SETPOS [0 -20] SETC 15
PUTSH 1 :NOAT
PUTSH 2 :STAFFLO
PUTSH 3 :STAFFHI
TELL 0 SETSH 2 TELL 1 SETSH 3
TELL 2 SETSH 1
END

TO BEEP :NOTE
SETENV 0 3
TOOT 0 :NOTE 10 20
END

TO PLAY :ST
IF :ST = "A [TYPE [\ c] SETY -20 OP 2
60.7]
IF :ST = "S [TYPE [\ d] SETY -15 OP 2
93.3]
IF :ST = "D [TYPE [\ e] SETY -10 OP 3
30]
IF :ST = "F [TYPE [\ f] SETY -5 OP 34
7.7]
IF :ST = "G [TYPE [\ g] SETY 0 OP 391
.1]
IF :ST = "H [TYPE [\ a] SETY 5 OP 440
]
IF :ST = "J [TYPE [\ b] SETY 10 OP 49
5]
IF :ST = "K [TYPE [\ C] SETY 15 OP 52
1.5]
IF :ST = "L [TYPE [\ D] SETY 20 OP 58
6.6]
IF :ST = "; [TYPE [\ E] SETY 25 OP 66
0]
IF :ST = "+ [TYPE [\ F] SETY 30 OP 69
5.4]
IF :ST = "* [TYPE [\ G] SETY 35 OP 78
2.2]
IF :ST = CHAR 32 [TYPE [\ ] SETY 0 OP
 50000]
OP 50000
END

TO BEEP2 :NOTE
SETENV 0 3
TOOT 0 :NOTE 10 20
TOOT 1 :NOTE / 2 8 20
END

TO REMEM :ST
MAKE "LINE LPUT :ST :LINE
END

TO REPLAY :LINE
IF EMPTYP :LINE [STOP]
IF :CHOI = "Y [BEEP2 PLAY FIRST :LINE
] [BEEP PLAY FIRST :LINE]
REPLAY BF :LINE
END

TO START
SETUP
BEEP2 PLAY "A
DIRECTIONS
GET.NOTE
END

TO CLEANUP
CT
PR [C\=CLEANUP,Y\=START OVER,R\=REPLA
Y]
MAKE "LINE []
END

TO GET.NOTE
.DEPOSIT 731 255
MAKE "ST RC
IF :ST = "C [CLEANUP GET.NOTE]
IF :ST = "Y [START]
IF :ST = "R [REPLAY :LINE GET.NOTE]
REMEM :ST
IF :CHOI = "Y [BEEP2 PLAY :ST] [BEEP
PLAY :ST]
GET.NOTE
END

MAKE "ST "R
MAKE "STAFFHI [8 12 10 9 9 9 9 9 9 9
9 9 9 10 12 8]
MAKE "STAFFLO [8 24 40 72 136 136 136
 138 141 141 139 137 137 73 42 28]
MAKE "NOAT [0 7 15 12 8 8 8 8 8 8 104
 248 248 240 96 0]
MAKE "CHOI "Y
MAKE "LINE [S D F \  F G H \  H J K \
  K K K]
```

assembly language

TURBO TYPO II

Article on page 43.

LISTING 1

```
WB 32000 REM TYPO II BY ANDY BARTON
CW 32015 REM TURBO TYPO REV. BY J.D. McLA
   UGHLIN
UP 32016 REM (c). 1985, ANTIC PUBLISHING
GI 32025 DIM B$(75)
ZS 32026 B$="[illegible]
   [illegible]
   [illegible]"
PX 32150 ANS=USR(ADR(B$),ADR(LINE$),LEN(L
   INE$)):ANS=PEEK(1789)+256*PEEK(1790)+6
   5536*PEEK(1791)
```

LISTING 2

```
SW 10 REM CREATE LINE 32026 FOR TURBO TYP
   O
CH 20 REM BY J.D. McLAUGHLIN
FW 30 REM (c) 1985, ANTIC PUBLISHING
LS 40 GRAPHICS 0
RT 50 RESTORE :POSITION 2,6:? "32026 B$="
   ;CHR$(34);
IE 60 POKE 766,1
GZ 70 FOR I=1 TO 71:READ A
HG 80 ? CHR$(A);
GG 90 NEXT I:POKE 766,0
GL 100 POSITION 2,11:? "CONT":POSITION 2,
   2:POKE 842,13:STOP
WR 110 POKE 842,12:? "LINE 32026 HAS BEEN
    CREATED":END
FL 120 DATA 169,1,133,203,104,104,133,205
   ,104,133,204,104,104,133,207,169,0,141
   ,253,6,141,254,6,141
PO 130 DATA 255,6,160,0,165,203,133,208,1
   77,204,133,206,24,165,206,109,253,6,14
   1,253,6,144,11,238
KW 140 DATA 254,6,173,254,6,208,3,238,255
   ,6,198,208,208,230,200,230,203,198,207
   ,208,215,96,34
```

LISTING 3

```
05 ; TURBO TYPO
06 ; BY J.D. McLAUGHLIN
07 ; (c) 1985, ANTIC PUBLISHING
10         *=  $0600
20 BASIC1 =   1789      ;FIRST
30 BASIC2 =   1790      ;SECOND AND
40 BASIC3 =   1791      ;THIRD BYTES OF A
NS FROM BASIC PROGRAM
50         LDA #1
60         STA $CB      ;ESTABLISH COUNTE
R (TYPOII'S VARIABLE C)
70         PLA          ;DISCARD #OF BYTE
S PASSED
80         PLA
90         STA $CD      ;HI BYTE OF ADR(L
INE$)
0100       PLA
0110       STA $CC      ;LO BYTE OF ADR(L
INE$)
0120       PLA          ;BASIC ALWAYS PAS
SES A 0 HERE AS HI BYTE OF LEN(LINE$)
0130       PLA
0140       STA $CF      ;LO BYTE OF LEN(L
INE$)
0150       LDA #0
0160       STA BASIC1   ;ZERO ALL
0170       STA BASIC2   ;BYTES OF
0180       STA BASIC3   ;ANS
0190       LDY #0       ;SET COUNTER
0200 LOOPA LDA $CB      ;GET VALUE OF 'C'
0210       STA $D0      ;USE AS A COUNTER
0220       LDA ($CC),Y  ;GET CHARACTER FR
OM LINE$
0230       STA $CE
0240 LOOPB CLC          ;LOOPB 'MULTIPLIE
S' CHAR. BY 'C'
0250       LDA $CE
0260       ADC BASIC1
0270       STA BASIC1
0280       BCC ONW
0290       INC BASIC2
0300       LDA BASIC2
0310       BNE ONW
0320       INC BASIC3
0330 ONW   DEC $D0
0340       BNE LOOPB
0350       INY
0360       INC $CB      ;SAME AS C=C+1
0370       DEC $CF      ;REDUCE NUMBER OF
 CHAR.S REMAINING
0380       BNE LOOPA    ;ALL CHARS. PROCE
SSED?
0390       RTS          ;YES,THEN RETURN
```

G.U.P.

continued from page 68

LISTING 2

Don't type the TYPO II Codes!

```
XU 10 REM GRAPHICS UTILITY PACKAGE DEMO
MG 20 REM BY DAREK MIHOCKA
FW 30 REM (c) 1985, ANTIC PUBLISHING
JP 40 START=16384
VE 42 IF PEEK(START)=76 THEN 50
ZA 44 POKE 710,66:? "↖↓    G.U.P. Machine
    Language routines  are not in memory!
   "
FL 46 ? :? "     This demo cannot RUN with
   out theseroutines.  See article for ad
   ditional information.":END
PD 50 RANDOM=PEEK(START+3)+256*PEEK(START
   +4)
DV 60 BOX=PEEK(START+5)+256*PEEK(START+6)

BE 70 LET DRAWTO=PEEK(START+7)+256*PEEK(S
   TART+8)
MJ 80 LINE=PEEK(START+9)+256*PEEK(START+1
   0)
SZ 90 CIRCLE=PEEK(START+11)+256*PEEK(STAR
   T+12)
IJ 100 LET COLOR=PEEK(START+13)+256*PEEK(
   START+14)
TD 110 LET GRAPHICS=PEEK(START+15)+256*PE
   EK(START+16)
IN 120 C128=PEEK(START+17)+256*PEEK(START
   +18)
RR 130 CIRF=PEEK(START+19)+256*PEEK(START
   +20)
XF 140 TEXT=PEEK(START+21)+256*PEEK(START
   +22)
ZG 150 LET PLOT=PEEK(START+23)+256*PEEK(S
   TART+24)
OE 160 LET SET=PEEK(START+25)+256*PEEK(ST
   ART+26)
PG 170 LET SETCOLOR=PEEK(START+27)+256*PE
   EK(START+28)
TC 180 HOROFFSET=208
UN 190 REM BEGIN MAIN PROGRAM
UJ 200 GOSUB 910
XT 210 POKE CIRF,1
GE 220 A=USR(GRAPHICS,15):POKE 752,255
SI 230 ? "↖The same routine in GR.15 with
    G.U.P."
QT 240 ? "using multiple parameters for s
   peed.":A=USR(SET,85,85,85,85)
BI 250 FOR T=0 TO 79 STEP 3:T2=T/2:U=159-
   T:U2=159-T2
LN 260 A=USR(LINE,T,T2,U,U2,T,U2,U,T2,T,T
   2)
SM 270 NEXT T:POKE 752,255
FU 280 ? "↖Twice the pixels in half the t
   ime..."
YZ 290 ? "How about some multi-color disp
   lays?"
DQ 300 A=USR(SET,164,29,134,165):GOSUB 11
   60:FOR T=79 TO 0 STEP -1
AC 310 A=USR(LINE,T,T,T,159-T)
FG 320 A=USR(LINE,159-T,159-T,159-T,T)
KF 330 NEXT T
RE 340 GOSUB 1100
GL 350 A=USR(GRAPHICS,15):POKE 752,255
YE 360 ? "↖And now, the Super Fast CIRCLE
   ..."
DJ 370 ? "30 CIRCLES DRAWN IN 4.8 SECONDS
          (6 CIRCLES PER SECOND!)"
XX 380 FOR Y=1 TO 30
FR 390 A=USR(RANDOM)
QX 400 A=USR(CIRCLE,60+Y,60+Y,60)
MA 410 NEXT Y
RB 420 GOSUB 1100
PS 430 A=USR(GRAPHICS,15)
VF 440 POKE 752,255:? "More fast circles.
   .."
DR 450 FOR R=80 TO 8 STEP -4
YZ 460 X=PEEK(20):A=USR(SET,X,X,X,X):A=US
   R(CIRCLE,79,79,R)
ES 470 A=USR(SET,0,0,0,0):A=USR(CIRCLE,79
   ,79,R-2)
JW 480 NEXT R
RP 490 GOSUB 1100
GD 500 A=USR(GRAPHICS,15):POKE 752,255
YI 510 ? "↖ And now, some more LINE drawi
   ng..."
XV 520 FOR T=0 TO 70
UE 530 GOSUB 1160
KK 540 A=USR(LINE,T,T,T+79,T,T+79,T+79,T,
   T+79,T,T)
LY 550 A=USR(SET,0,0,0,0):T=T+1
KO 560 A=USR(LINE,T,T,T+79,T,T+79,T+79,T,
   T+79,T,T)
KP 570 NEXT T
RO 580 GOSUB 1100
QN 590 A=USR(GRAPHICS,15):POKE 752,255:?
   "↖Of course, circles can be left empty
   "
TZ 600 GOSUB 1160
XI 610 POKE CIRF,0
DY 620 FOR T=0 TO 50 STEP 2
FF 630 A=USR(SET,170,170,170,170)
GS 640 A=USR(CIRCLE,T+45,79,60)
HB 650 A=USR(SET,0,0,0,0)
HR 660 A=USR(CIRCLE,T+46,79,60)
KQ 670 NEXT T
GS 680 GOSUB 1100:A=USR(GRAPHICS,15):POKE
    752,255
JE 690 ? "↖              THE BOX..."
WK 700 FOR T=0 TO 70:U=3*T:A=USR(SET,U,U,
   U,U)
HG 710 A=USR(BOX,T,T,159-T,159-T):NEXT T
RE 720 GOSUB 1100
QO 730 GRAPHICS 8:POKE 752,255
DN 740 ? "↖How about mixed text and graph
   ics???"
TT 750 FOR T=0 TO 3:POKE COLOR+T,255:NEXT
    T
XK 760 FOR T=1 TO 50
KZ 770 A=USR(CIRCLE,PEEK(53770)/2+16,PEEK
   (53770)/2+16,9+PEEK(53770)/40)
MF 780 POKE CIRF,PEEK(53770)/256
UD 790 A=USR(TEXT,34,8,ADR("This complete
   s the demo"),23)
HE 800 A=USR(TEXT,30,24,ADR("of the GRAPH
   ICS UTILITIES"),25)
KG 810 NEXT T
DD 820 POKE TYPE,0:A=USR(TEXT,32,8,ADR("
                                  "),33)
WR 830 A=USR(TEXT,28,32,ADR("
                  "),26)
JK 840 POKE TYPE,255:A=USR(TEXT,34,8,ADR(
   "This completes the demo"),23)
HO 850 A=USR(TEXT,30,24,ADR("of the GRAPH
   ICS UTILITIES"),25)
BE 860 A=USR(C128,120,5)
CA 870 A=USR(C128,120,6)
RR 880 GOSUB 1100
VD 890 GRAPHICS 18:POKE 712,48:? #6:? #6;
   "THANKS FOR WATCHING":? #6:? #6:? #6;"
         the"
DQ 900 ? #6;"  graphics utility        pa
   ckage":X=USR(C128,400,5):END
QS 910 GOSUB 1000
BY 920 A=USR(GRAPHICS,15):POKE 87,7:REM M
   AKE BASIC THINK IT'S IN GR.15.
IU 930 COLOR 1:POKE 752,255
DO 940 ? "Limited by the O.S., BASIC can
   only   use 160x80 resolution in GRAPHI
   CS 15. It is also slow."
RI 950 FOR T=0 TO 79 STEP 3:T2=T/4:U=159-
   T:U2=79-T2
FC 960 PLOT T,T2:DRAWTO U,U2:DRAWTO T,U2:
   DRAWTO U,T2:DRAWTO T,T2
KT 970 NEXT T
RS 980 GOSUB 1100
ZX 990 RETURN
ME 1000 GRAPHICS 18:? #6:? #6;"  IN THE B
   EGINNING"
NN 1010 ? #6:? #6;"    THE ATARI HAD "
VO 1020 ? #6:? #6;"   PLOT AND DRAWTO "
NC 1030 FOR T=1 TO 500:NEXT T
EG 1040 GRAPHICS 1:? #6:? #6;"NOW, THANKS
    TO G.U.P"
NL 1050 ? #6:? #6;" THE ATARI ALSO HAS"
JM 1060 ? #6:? #6;" BOX, CIRCLE,  AND"
JW 1070 ? #6:? #6;" 16 GRAPHICS MODES"
XU 1080 GOSUB 1100
BA 1090 RETURN
EH 1100 POKE 752,255
UJ 1110 ? "↖↓  Press a key to continue de
   mo...."
FZ 1120 POKE 764,255
TN 1130 IF PEEK(764)=255 THEN 1130
GF 1140 POKE 764,255
AQ 1150 RETURN
NH 1160 A=USR(RANDOM):RETURN
```

product reviews

COLORTONE KEYBOARD

(Waveform Corp.)
Protecto Enterprizes
Box 550
Barrington, Il 60010
(312) 382-5244

Reviewed by Nat Friedland

At $49, the Colortone Keyboard by Waveform Corp. is not a tool for professional musicians. But nevertheless it is an intriguing Atari music device that puts a surprising amount of musical power and feedback literally at your fingertips.

The Colortone is a recent conversion from a fairly successful Commodore 64 product. As a matter of fact, **Antic** received its copy just one day after programmer Russ Karras delivered the conversion software to Waveform. Actually our keyboard's converter hookup is a prototype that dangles exposed on a spliced cable.

Distribution for this product is just getting set up, but one source you should be able to obtain it from is the Protecto mail-order house whose address is shown above. (We recommend that you verify availability by phone before mailing Protecto a check.)

The Colortone is a membrane keyboard, after the fashion of the old Atari 400. It's laid over a sturdy plastic base, but looks a lot like those cardboard piano keyboards that kids in group piano classes used for practicing their fingering.

Once you select your choices from the function keys at the top of the board, you will probably wind up playing mostly on the Color Harp strip above the regular piano-key diagram. The Color Harp gives you only the notes in the scale you have chosen, so you can fool around without the risk of hitting a drastically wrong note.

The Colortone software does a lot of interesting things. You get a choice of seven well done pop/rock two-voice accompaniments or a simple metronome beat. Some of the most common and effective rock patterns are utilized, boogie, ballad, etc.

You have eight instrumental voices to choose from and a variety of musical scales. You can also adjust the speed, pause, and listen to playback of your solo with or without saving it to disk.

As you are playing, the software writes out the musical notation of your solo, showing it above the bass accompaniment notes. At the same time, it also tracks your fingering with red lights over a piano key display.

It is easy to sound pretty good as you play along with the accompaniment by tapping or even rubbing your fingers over the color harp. You can never be drastically out of tune, although you are able to experiment with interestingly dissonant effects if you wish.

A lot of your noodling will sound like the background music from your favorite videogames.

Where Colortone Keyboard eventually shows its limitations is its lack of precision at reflecting musical ideas you are consciously trying to play. It's often nearly impossible to find a specific note you're looking for in the color harp section. And the membrane sensors on the piano diagram don't respond fast enough to pick up speedy passages.

Eventually it seems that no matter what you try to do, you find yourself repeating a consistent pattern of chromatic eighth-note scales as the accompaniments pump steadily along.

ADVERTISERS

This is provided as a convenience and as a courtesy to advertisers. **ANTIC** does not guarantee accuracy or comprehensiveness.

SHOPPER'S GUIDE

atari service center

Service Centers, Retailers, to get your listing in Antic call (415)661-3400

ALABAMA

VIDEO REPAIR CO.
2009 CENTERPOINT RD.
BIRMINGHAM
205-854-5212

RAINBOW CITY SERVICE CENTER
244 RAINBOW PLAZA
GADSEN
205-442-6810

C & R ELECTRONICS
704 HOLCOMBE AVE.
MOBILE
205-473-3030

BUSINESS SERVICES
2828 CHESTNUT ST.
MONTGOMERY
205-834-2290

ARIZONA

RICK'S TV & APPLIANCE
1104 E. DEUCE OF CLUBS
SHOW LOW
602-537-7625

CALIFORNIA

LEARNING TREE COMPUTER CTR.
2441 N. TUSTIN SUITE BCD
SANTA ANA
714-667-1575

COMPUTER SUPPORT SERVICE
52 S. LINDEN AVE. #1
SOUTH SAN FRANCISCO
415-589-9800

D & G COMPUTERS
4156 MANZANITA AVE. #200
CARMICHAEL
916-485-7779

SAN JOSE COMPUTER
1844-E ALMADEN RD.
SAN JOSE
408-723-2025

B & C COMPUTERVISION
3400 EL CAMINO REAL
SANTA CLARA
408-554-0666

ATCOM COMPUTERS
1421 THOUSAND OAKS BLVD.
THOUSAND OAKS
805-497-1220

AUTHORIZED COMPUTER SERVICE
951 W. FOOTHILL BLVD.
UPLAND
714-985-2101

TESTEK
7224 VALJEAN AVE.
VAN NUYS
818-786-6890

COMPUTER JUNCTION, INC.
15000 7TH ST. SUITE 214
VICTORVILLE
619-245-3622

COLORADO

AMERICAN TELEVISION SERVICE CO
1226 W. LITTLETON BLVD.
LITTLETON
303-795-2040

LOOKING GLASS MICROPRODUCTS
4233 WEST EISENHOWER
LOVELAND
303-669-2681

FLORIDA

R & S ELECTRONICS, INC.
3245 W. MCNAB RD.
FT. LAUDERDALE
305-979-6763

BOURQUE'S ELECTRONIC SERVICE
180 RICHPIEN RD.
FT. WALTON BEACH
904-862-3346

MR. SOFTWARE
101 HOLLYWOOD FASHION CENTER
HOLLYWOOD
305-981-9090

ENTERTAINMENT SERVICES ELECT.
811 EDGEWOOD AVE. S.
JACKSONVILLE
904-786-1305

COMPUTER IMAGE
10061 SUNSET DR.
MIAMI
305-271-1224

COMP-U-PHONE
6160 EDGEWATER DR.
SUITE E.
ORLANDO
305-291-1712

GEORGIA

GOODWIN'S ELECTRONICS
JUNCTION SHOPPING CENTER
ALBANY
912-435-9605

NORMAN'S ELECTRONICS, INC.
4014 PEACHTREE RD. NE
ATLANTA
404-237-3349

ELECTRONIC SPECIALISTS
1685 EDNA PLACE
MACON
912-742-5628

RADIOS AND MICROCOMPUTERS INC.
3833A WASHINGTON RD
MARTINEZ
404-863-9071

HARRIS TV
1 E. MONTGOMERY CROSSROAD
SAVANNAH
912-927-2084

HAWAII

KONA COMPUTER
75-5706 HANAMA PL. #107
KAILUA-KONA
808-329-8574

ILLINOIS

DIGITAL WORLD, INC.
711 ARMY TRAIL RD.
ADDISON
312-543-9000

OMEGA ENTERPRISES
7823 N. 2ND ST.
ROCKFORD
815-282-1477

INDIANA

WRIGHT ELECTRONICS
614 N MAIN ST.
EVANSVILLE
812-423-2845

CITIZEN'S TV & VIDEO
827 W. GLENPARK AVE.
GRIFFITH
219-924-0047

THE COMPUTER CORNER
7101 BROADWAY
MERRILLVILLE
219-738-3282

KANSAS

MIDWEST APPLIANCE SERVICE
METCALF SOUTH MALL
OVERLAND PARK
913-341-6688

KENTUCKY

FACTORY ELECTRONICS
2422 PALUMBO DR.
LEXINGTON
606-269-7341

VANOVER TV HOSPITAL
2027 CUMBERLAND AVE.
MIDDLESBORO
606-248-3827

LOUISIANA

COMPUTER ELECTRONICS
1955 DALLAS DR.
BATON ROUGE
504-924-8066

MASSACHUSETTS

CUSTOM ELECTRONICS
238 EXCHANGE ST.
CHICOPEE
413-592-4761

BEACON TV-ELECTRONICS
8 TILESTON ST.
EVERETT
617-389-8600

CONDOR SERVICE
372 SOUTH MAIN ST.
SHARON
617-784-2382

ROCOM, INC.
184 MAIN ST.
WAREHAM
617-295-2542

MARYLAND

NATIONAL BUSINESS & SECURITY
8639 LOCH RAVEN BLVD.
BALTIMORE
301-665-8870

EDGEWOOD TV & AUDIO
4932 EDGEWOOD RD.
COLLEGE PARK
301-441-9116

CROFTON TV & VIDEO SERVICE
2217 DEFENSE HWY
CROFTON
301-721-1700

COMPUVISION COMPUTER CENTER
6445 BURWOOD PL.
GLEN BURNIE
301-850-4055

ADVANCED COMPUTER SERVICE CO.
287 E. GREEN ST.
WESTMINSTER
301-876-8202

MAINE

AUTOMATIC SERVICES
22 TARGET INDUSTRIAL CIRCLE
BANGOR
207-942-6769

J.D. ELECTRONICS
385 STEVENS AVE.
PORTLAND
207-775-1411

MICHIGAN

FUTURE DIRECTIONS
1520 N. VAN DYKE
BAD AXE
517-269-7211

THE FAMILY COMPUTER CENTER
3895 W. 12 MILE RD.
BERKLEY
313-543-0520

CHASE TRANSISTOR SERVICE
521 LEONARD ST. NW
GRAND RAPIDS
616-454-9000

ABL ELECTRONIC SERVICE, INC.
32 E. 14 MILE ROAD
MADISON HEIGHTS
313-588-6663

SOLID STATE SERVICE
548 SHATTUCK RD.
SAGINAW
517-752-0606

MINNESOTA

PHILIPS COMMUNICATIONS & TV
748 N.E. HARDING ST.
MINNEAPOLIS
612-378-7200

USER FRIENDLY COMPUTER
8465 PLAZA BLVD.
SPRING LAKE PARK
612-786-8181

MISSOURI

SOUTHLAND ELECTRONICS
22 E. OLIVE DR.
AURORA
417-678-4623

MIDWEST APPLIANCE SERVICE
METRO NORTH MALL
KANSAS CITY
816-436-7010

COMMUNITY SOUND & VIDEO
1834 S. STEWART
SPRINGFIELD
417-887-3391

A & E ELECTRONICS CORP
2001 BIG BEND BLVD.
ST. LOUIS
314-645-7733

J & S VIDEO VISIONS
1051 WASHINGTON SQUARE CENTER
WASHINGTON
314-239-2677

MISSISSIPPI

ELECTRONIC SERVICES
2315 25TH AVE
GULFPORT
601-863-3772

NORTH CAROLINA

SOUTHERN PHOTO TECHNICAL SVCE
2610 SOUTH BLVD.
CHARLOTTE
704-523-0012

NEW JERSEY

DEPENDABLE PARTS, INC.
168 MAIN ST.
CHATHAM
201-635-5888

S R DATA
2141 WOODBRIDGE AVE.
EDISON
201-985-5017

ADVANCED ELECTRONICS
VILLAGE SHOPPING CENTER
RT.206
FLANDERS
201-584-1252

TELSAR ELECTRONIC SERVICES, INC
829 STONE RD.
LAUREL SPRINGS
609-783-8500

VIDEO ELECTRONICS COMPUTER SVC
1418 ROSELLE ST.
LINDEN
201-925-1418

VIDEO CONNECTION OF SOMERSET
900 EASTON AVE.
SOMERSET
201-545-8733

K & S ELECTRONICS
119 HAMILTON BLVD.
SOUTH PLAINFIELD
201-755-4204

NEW YORK

ISLAND VIDEO COMPUTER SERVICE
35 MIDDLE COUNTRY RD.
CORAM
516-736-1001

AARDVARK ELECTRONIC SERVICE
44 CASTLE ST.
GENEVA
315-789-5295

LONG ISLAND COMPUTER GENERAL
103 ATLANTIC AVE.
LYNBROOK
516-887-1500

ABC ELECTRONICS SERVICE CO.
392 THURSTON RD.
ROCHESTER
716-328-1840

OHIO

ARJAY MICRO
1385 BETHEL RD.
COLUMBUS
614-459-4219

STATION FUN TV, INC.
206 CLINTON ST.
DEFIANCE
419-782-8545

COMPUTER CREATIONS
424 E. STROOP RD.
KETTERING
513-294-0222

B AND G ELECTRONICS, INC.
15729 MADISON AVE.
LAKEWOOD
216-521-2855

ERIC MARTIN'S, INC.
5485 WARRENSVILLE CENTER RD.
MAPLE HGTS
216-663-2032

VIDEO COMPUTER WORLD, INC.
2223 WOODVILLE RD.
OREGON
419-691-7282

DODD CO.
7795 W. RIDGEWOOD DR.
PARMA
216-886-2828

OREGON

NORTHWEST COMPUTER SUPPORT,INC
10200 S.W. NIMBUS G-1
PORTLAND
503-684-3280

PENNSYLVANIA

KIBLER'S INDEPENDENT TV
526 FALLOWFIELD AVE.
CHARLEROI
412-483-7484

PARK T. MORROW, INC.
627 W. 26TH ST.
ERIE
814-455-7566

DEBUG BYTES COMPUTERS
662 PHILADELPHIA ST.
INDIANA
412-349-7290

GRUSS ELECTRONIC REPAIR
HILLS PLAZA
JOHNSTOWN
814-266-1395

TESCO, INC.
9237 ROOSEVELT BLVD.
PHILADELPHIA
215-677-5000

atari service center

NATIONAL TELEVISION SERVICE
5461-63 PENN AVE
PITTSBURGH
412-361-5400

BOYD TV
719 LANCASTER AVE
WAYNE
215-688-3727

RHODE ISLAND

VIDEO ENCOUNTERS
MAINES SHOPPING CENTER
WAKEFIELD
401-783-3460

SOUTH CAROLINA

ELECTRONIC SERVICE CO.
1736 DECKER BLVD
COLUMBIA
803-782-2705

COASTAL TV & APPLIANCE CO.
603 HWY 501
CONWAY
803-248-2686

SOUTH DAKOTA

TAYLOR AUDIO-VISUAL, INC.
1009 DAKOTA S.
HURON
605-352-3205

HOUSE OF TELEVISION
601 SOUTH DULUTH AVE.
SIOUX FALLS
605-338-9051

TENNESEE

HI-FI SERVICE CENTER
4608 HIXSON PIKE
CHATTANOOGA
615-877-6781

BILL'S T.V. SALES & SERVICE
3843-C DICKERSON RD.
NASHVILLE
615-865-5000

TEXAS

TV CENTER
202 S. WILLIS
ABILENE
915-677-1171

PACIFIC STEREO
525 113TH ST.
ARLINGTON
817-640-3094

LONGS/ENTRONIX
5800 MAPLE AVE.
DALLAS
214-358-3222

COMPUTER HOME, INC.
3548 KNICKERBOCKER
SAN ANGELO
915-944-9795

UTAH

INTERWEST ELECTRONICS CORP.
4091 SOUTH STATE ST.
SALT LAKE CITY
801-266-5301

VIRGINIA

SALEM COMPUTER CENTER
4034 PLANK RD.
FREDERICK
703-786-8126

VIDEO UNLIMITED SERVICE CENTER
1707 ROUTE 17
GRAFTON
804-898-5318

L & Y ELECTRONICS
13670 JEFFERSON DAVIS HWY.
WOODBRIDGE
703-494-3444

WASHINGTON

ON LINE COMPUTERS PLUS
13710 NE 20TH ST.
BELLEVUE
206-644-2080

BUTLER'S TV & COMPUTER SERVICE
28717 PACIFIC HWY SOUTH
FEDERAL WAY
206-941-9096

ARTICULATE SYSTEMS
E 9405 SPRAGUE AVE.
SERVICE CTR
SPOKANE
509-922-0255

JOHNSON'S TELEVISION SERVICE
N 4424 WALL
SPOKANE
509-327-9566

WISCONSIN

AUTHORIZED TV
810 NINTH ST.
GREEN BAY
414-499-4215

DAN'S CITY WIDE T.V.
1259 E. JOHNSON ST.
MADISON
608-255-4144

MISTER TV SERVICE
5455 W. BURLEIGH ST.
MILWAUKEE
414-873-2415

WEST VIRGINIA

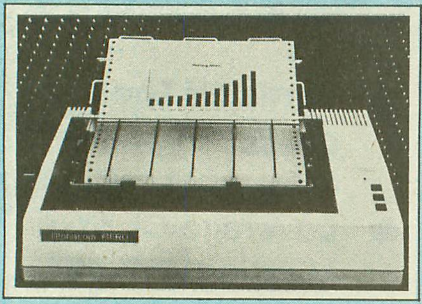
COMPUTERS PLUS, INC.
2077 CHARLESTON TOWN CENTER
CHARLESTON
304-342-4848

Next Month

COMPUTER CHALLENGES

- Battle Of The Chess Programs
- Coverage Of Atari Meeting At Antic
- Inside Strategic Simulations Inc.
- Another Mind-Boggling Cover By Okamoto

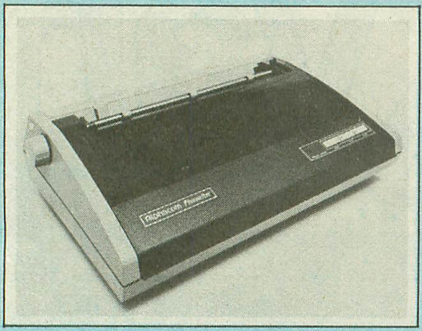
new products



AERO

(printer)
Alphacom
2323 South Bascom Ave.
Campbell, CA 95008
(408) 559–8000
$249.95

Described as an Epson-compatible dot-matrix printer, the Aero is rated at 130 cps. It uses a 6x9 matrix field, contains a 2K buffer and supports (all on the same line) superscript, subscript, double strike, enlarged and emphasized print and underlining.

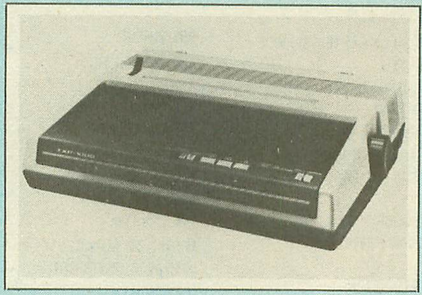


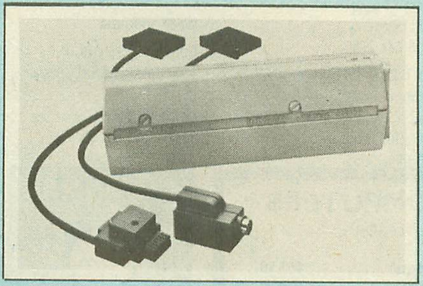
PLUS WRITER

(printer)
Alphacom
2323 SO. BASCOM Ave.
Campbell, CA 95008
(408) 559-8000
$399

Alphacom claims this new 20 character-per-second daisywheel printer is the perfect printer for your Atari. Plugging directly into your Atari, the PlusWriter emulates the Diablo 630 and uses standard ribbons and daisywheels. We are told that the printer supports superscript, subscript, boldface, proportional spacing and underlining.

New Products notices are compiled by the **Antic** staff from information provided by the products' manufacturers. Antic welcomes such submissions, but assumes no responsibility for the accuracy of these notices or the performance of the products listed.





GE 3–8100 PRINTER GE PRINTER INTERFACE MODULE

(printer and interface)

General Electric Company
Electronics Park
Syracuse, NY 13221
(315) 456–2446

$299.95 (printer)
$89.95 (interface)

GE's foray into the home computer market includes a dot-matrix printer capable of "letter quality" print at 25 cps or regular draft printing at 50 cps.

The interface will connect the printer directly to an Atari.

QUESTBUSTERS

(adventure newsletter)
202 Elgin Court
Wayne, PA 19087
$15/year
This monthly adventure game newsletter covers recent news, reviews, and the like. It is not Atari-specific.



WIRE TREE PLUS

(line filter)
Networx
203 Harrison Place
Brooklyn, NY 11237
(718) 821–7555
$99.95

This surge/spike filter not only protects electrical lines for computers and peripherals, but also protects two modular phone lines (providing exactly the kind of filtering and protection the Atari 1030 modem doesn't have).

Q-MODEM

(modem)
Quantum Microsystems Inc.
P.O. Box 179
Liverpool, NY 13088
(315) 451–7747
$149.95

A 300-baud direct connect modem with autodial, autoanswer and other advanced features, the Q-Modem plugs directly into your Atari and fits in the palm of your hand. It even comes with the famous QuanTerm software, which uses color displays, on-screen menus, a built-in editor, multiple buffers and so on.

Return the favor. When you call a manufacturer or supplier about a product you've seen advertised or otherwise mentioned in ANTIC, please tell them so. This will help us to continue to bring you the latest information about products that will make your Atari computer an even more valuable investment in the future. —ANTIC ED

Turbo charge your Atari with an Indus GT.™

Introducing the all-new Indus GT™ disk drive. The most advanced, most complete, most handsome disk drive in the world.

A flick of its "Power" switch can turn your Atari into a Ferrari.

Looks like a Ferrari.
The Indus GT is only 2.65" high. But under its front-loading front end is slimline engineering with a distinctive European-Gran flair.

Touch its LED-lit CommandPost™ function control AccuTouch™ buttons. Marvel at how responsive it makes every Atari home computer.

Drives like a Rolls.
Nestled into its soundproofed chassis is the quietest and most powerful disk drive power system money can buy. At top speed, it's virtually unhearable. Whisper quiet.

Flat out, the GT will drive your Atari track-to-track 0-39 in less than one second. And when you shift into SynchroMesh DataTransfer,™ you'll increase your Atari's baud rate an incredible 400%. (Faster than any other Atari system drive.)

And, included as standard equipment, each comes with the exclusive GT DrivingSystem™ of software programs. World-class word processing is a breeze with the GT Estate WordProcessor.™ And your dealer will describe the two additional programs that allow GT owners to accelerate their computer driving skills.

Also, the Indus GT is covered with the GT PortaCase.™ A stylish case that conveniently doubles as a 80-disk storage file.

Parks like a Beetle.
The GT's small, sleek, condensed size makes it easy to park.

So see and test drive the incredible new Indus GT at your nearest computer dealer soon.

The drive will be well worth it.



INDUS™

The all-new Indus GT Disk Drive.

The most advanced, most handsome disk drive in the world.



For additional information, call 818/882-9600.



Expand your Atari
...With peripherals from MPP

MicroPrint
Parallel Printer Interface

- Works with Atari, 400, 800, 600XL and 800XL
- Replaces Atari 850™ Interface Module
- Compatible with all software
- 5-foot cable with Centronics plug (compatible with Epson, NEC, Prowriter, etc.)
- Connects to serial bus on computer
- 2 Year Warranty



MPP-1000C
Modem

- Auto Answer / Auto Dial
- Direct Connect to Phone Line
- No Atari 850™ Interface Module Needed
- Includes AC Adapter / Power Supply
- Free CompuServe DemoPak™
- 1 Year Warranty
- Connects to Joystick Port
- Works on **ALL** Atari Computers

SOPHISTICATED SMART TERMINAL SOFTWARE ON CARTRIDGE

FEATURES:

- ☐ Supports XMODEM Protocol
- ☐ ASCII / ATASCII Translation
- ☐ Allows Transfer of Files Larger than Memory
- ☐ Upload / Download of Text and Programs
- ☐ 100% Machine Language
- ☐ Loads a 65 Column Screen Driver
- ☐ Multiple Buffers
- ☐ Off-Line Editing
- ☐ Variable Baud Rate
- ☐ Parity Options
- ☐ Full / Half Duplex

ONLY
$149.95

MICROBITS PERIPHERAL PRODUCTS
225 Third Avenue, SW • Albany, OR 97321
ORDERS: 1-800-624-7532
CUSTOMER SERVICE: 1-503-967-9075

Atari and CompuServe DemoPak are trademarks of Atari Corp. and CompuServe.